REVISTA PARA USUARIOS DE

# Drean commodore

Nº 4 ₳ 2.00 REP. ARGENTINA



√ Compútelo a su favor

# En Argecint la Drean Commodore trae una tecla más: el respaldo.




DATASETTE

COMMODORE 16

JOYSTICK

El respaldo, el service y el asesoramiento que puede ofrecer una empresa con 15 años dedicados a la comercialización de todo lo que tenga que ver con el mundo de la computación. Argecint, El Súper Todo de Computación. Que pone a su alcance la Drean Commodore, con la línea más completa de accesorios, programas, periféricos... y con la financiación más cómoda de plaza y donde siempre encontrará un asesoramiento de expertos y una atención de amigos.

PREMIUM

ARGECINT




COMPUTADORES
PERIFERICOS
MAGNETICOS
MUEBLES
CINTAS
CASSETTES
ACCESORIOS
SUMINISTROS
FORMULARIOS
LAB. TECNICO
CURSOS DE COMPUTACION

**Drean Commodore y Argecint, la combinación perfecta.**

**PLANES UNICOS DE FINANCIACION - TARJETAS DE CREDITO**

**Casa Matriz:** VENTURA BOSCH 7065 - Tel.: 641-0507/4882/3051 TELEX 17312 (ERSA) C.C. 8 Suc. 8 (1408) Cap. Fed.
**Casa Central:** AV. DE MAYO 1402 - Tel.: 37-4631 - Cap. Fed.
**Agencia Trust:** CARLOS PELLEGRINI Y CORRIENTES - Tel.: 35-5018/5029/0344 - Cap. Fed.
**Agencia Belgrano:** COMPUMARKET - AV. CABILDO 2868/71 - Tel.: 785-5241/4689 - Cap. Fed.
**Agencia Flores:** TRUST JOYERO - AV. RIVADAVIA 9887 - Tel.: 634-4639 - Cap. Fed.
**Agencia Avellaneda:** HIJOS DE G. ROSSI - AV. MITRE 882 - Tel. 201-5658 - Bs. As.
**Sucursal Liniers:** AV. RIVADAVIA 11332 (1408) - Tel.: 641-3088 - Cap. Fed.
**Agencia Litoral:** PEATONAL SAN MARTIN 2455 - Loc. 38 (3000) - Tel.: 25459 - STA. FE
**Agencia Barrio Norte:** AV. SANTA FE 2546 - Tel.: 84-8870 - Cap. Fed.
**Agencia Computer Beach:** AV. J.C. CHIOZZA 2870 (6587) - Tel.: 291 - SAN BERNARDO - BS. AS.

# SUMARIO

## NOTAS TECNICAS



## PROGRAMAS

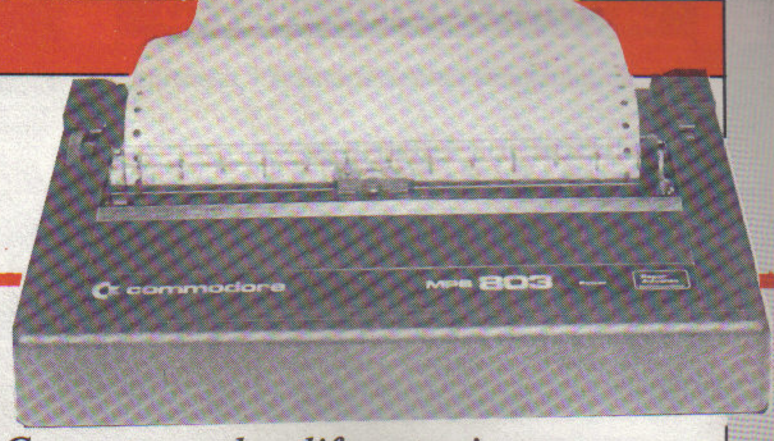


## REVISION DE SOFTWARE

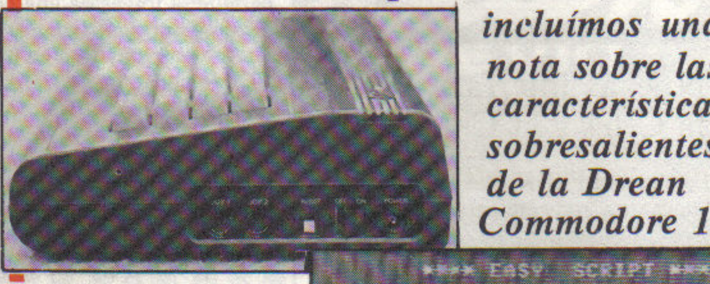


## SECCIONES FIJAS






*Comentamos las diferentes impresoras que hay en nuestro mercado para la Drean Commodore 64.*



*Para los que recién se inician, incluímos una nota sobre las características sobresalientes de la Drean Commodore 16*

```
**** EASY  SCRIPT ****

(C) PRECISION SOFTWARE LTD. 1982

ENTER TEXT WIDTH (40-240) COLS ? 80
(D)ISK  OR  (T)APE ? D
PRINTER TYPE (0-4) ? 0
0=CBM
1=MX80
2=SPINWRITER
3=QUME/DIABLO/3300
4=OTHER
```

*Calificamos dos buenos utilitarios y dos divertidos juegos.*


**Drean commodore** AÑO 1 Nº 4 MARZO DE 1986

**Director General**
Ernesto del Castillo
**Director Editorial**
Cristian Pusso
**Director Periodístico**
Fernando Flores

**Director Financiero**
Javier Campos Malbrán
**Arte y Diagramación**
Fernando Amengual
**Coordinador**
Ariel Testori

**Redacción**
Cristian Parodi
**Departamento de Avisos**
Oscar Devoto
**Departamento de Publicidad**
Guillermo Gònzalez Aldanur

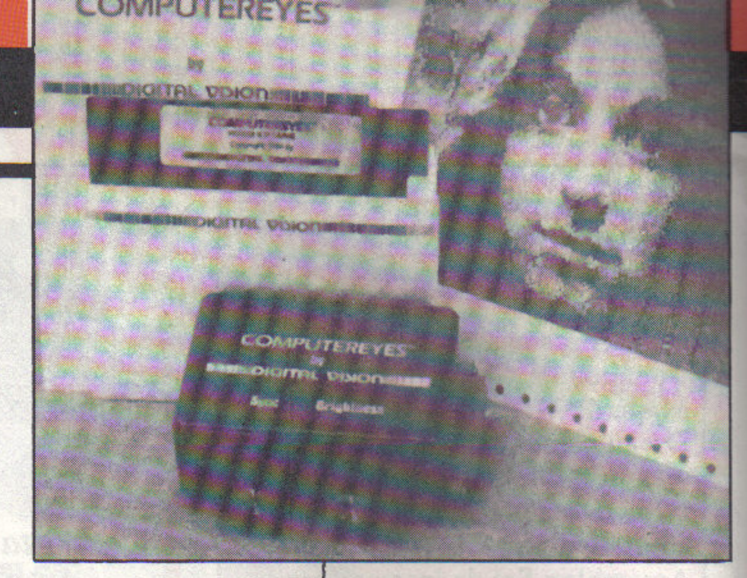
Drean Commodore es una Revista mensual editada por Editorial PROEDI S.A., Cerrito 1320, 1º Piso, Buenos Aires, Te.: 42-9681/9. Registro Nacional de la Propiedad Intelectual: E.T. M. registrada.



Precio de este ejemplar: ₳ 2.
Impresión: Calcotam. Fotocromo tapa: Columbia. Fotocomposición: Van Waveren.
Los ejemplares atrasados se venderán al precio del último número en circulación.



Distribuidor en Capital: Martino, Juan de Garay 358, P.B. Capital. Distribuidor interior: DGP: Hipólito Yrigoyen 1450, Capital Federal. T.E. 38-9266/9800




## Salieron los Cartridge

Están disponibles en el mercado nacional los nuevos Cartridge desarrollados por la empresa PEEK. Como habíamos adelantado en anteriores notas, PEEK dispone de la tecnología necesaria para pasar cualquier programa de hasta 32 Kb de longitud a este dispositivo. Este se conecta en la entrada especial para cartridge que tienen las Drean Commodore 64 y 16. No se requiere de datassette o drive.
Una de las características sobresalientes de este tipo de "periféricos" es que los programas pueden ser ejecutados directamente; es decir que no hace falta cargalos a memoria.

## Nuevas interfaces para la C-64 y la C-16

La empresa PEEK, que suministra hardware y software para los equipos Drean Commodore, ha lanzado en nuestro mercado tres nuevos periféricos. Se trata del datassette 1530-1531, el joystick P-8501 y el drive 1541.
El primero, modelo 1530, es compatible con la Drean Commodore 64, mientras que el 1531 fue diseñado para la 16. No necesitan alimentación de energía externa ya que ésta es suministrada por la computadora.
La conexión se realiza en el port que dispone cada uno de los equipos para tal fin. Cada una de ellas viene acompañada por su correspondiente manual en donde se nos explica la conveniencia de tener alejado el datassette del receptor de T.V., el procedimiento de conexión del equipo, la operación paso por paso, cómo desmagnetizar los cabezales del equipo y cómo efectuar su correcto mantenimiento y, en el final, cómo almacenar y recuperar información y trabajar con archivos.
El joystick P-8501 es una robusta, fiable y sólida palanca de mandos desarrollada para la Drean Commodore 64 y 16. Dispone de un disparador que actúa. prácticamente al tacto. Los movimientos se logran con sólo desplazar levemente el mando hacia la posición deseada. Su vida útil es mucho más alta que la de otros joysticks.
La unidad de disco 1541 es otra de las novedades. Es totalmente compatible con la C-64 y la C-16. Su capacidad de almacenamiento es de aproximadamente 175 Kb por diskette (éstos son de 5-1/4" simple lado/simple densidad). La unidad es del tipo "inteligente", ya que dispone de un microprocesador 6502 que se encarga del funcionamiento de la unidad. Otra de las ventajas es que nos permite trabajar con archivos relativos.
Un extenso manual acompaña al disk drive 1541 donde se explica, entre otras cosas, conexionado del equipo, operación, comandos permitidos para el uso de éste, etc.

## Fotografía con la C-64

La firma norteamericana Digital Vision ha desarrollado una interface que permite, a través de una cámara de video común, imprimir una determinada imagen en pantalla o en la impresora. Esta se constituye por diferentes matices de grises. Básicamente el funcionamiento de la interface consiste en un barrido lento de la imagen emitida por la cámara. Luego ésta es convertida de una señal analógica a una señal digital (constituídas por 0 y 1) a través de un proceso que se conoce con el nombre de muestreo. Finalmente estas series de 1 y 0 son procesados por la C-64 imprimidas en pantalla o impresora. Las aplicaciones de este producto son diversas. Estas pueden ser seguridad, arte gráfico, control industrial, reconocimiento de imágenes, entretenimientos, educación, robótica, inteligencia artificial, etc. Como mencionamos anteriormente acompañan a la interface el software que permite, por ejemplo, almacenar las imágenes capturadas por la cámara en el diskett, posibilidad de reducir la imagen, etc.



Drean commodore

C 16 y C 64

con:

Dreanplan

DE AHORRO PREVIO




en:

**20 CUOTAS SIN INTERES**

MOD. C16 20 CUOTAS DE ₳ 13,72
MOD. C64 20 CUOTAS DE ₳ 21,84

Administra:

PLAN CONFORT HOGAR S.A.

DE AHORRO PARA FINES DETERMINADOS

LUIS SAENZ PEÑA 310 - 5º PISO (1110) CAP. FED. Tel.: 37-1765 - 38-5812.

de la Cámara Argentina de Sociedades Administradores de Ahorro y Préstamo para Fines Determinados



# CONTABILIDAD Y FINANZAS DOMESTICAS

***Otra de las posibles utilizaciones de la Drean Commodore 64.***

La mayoría de la familias no tienen suficiente dinero como para contratar los servicios de un contador, pero desgraciadamente todavía tienen que archivar todo tipo de complicados documentos financieros y contables. Probablemente el objeto más popular para archivar los asuntos financieros es una caja, la cual suele estar bastante llena. Cada año, a la hora de hacer la declaración de impuestos, millones hacen la promesa de que el próximo año todo será diferente: no más cajas!. Se comprometen, además, a organizar la documentación de tal manera que la búsqueda de un determinado documento no lleve más de unos pocos segundos. Todos sabemos que el año llega rápidamente y que la organización sigue siendo un desastre. Las últimas semanas se dedican a buscar por todas partes los recibos, talonarios anulados, documentos de seguro o del médico, los cuales estaban ya la semana pasada en el cajón del escritorio.
La organización tan ansiada llega cuando las promesas se cambian por una pequeña pero potente computadora hogareña juntamente con los programas utilitarios respectivos. No obstante, la organización de los datos para su declaración de impuestos, no es la única área donde la computadora personal puede servir en beneficio de la familia. Las aplicaciones generales son:

## Gestión de los asuntos bancarios:

Cada mes, millones de personas se ocupan de sus talonarios anulados, talones y extractos de las transacciones bancarias. Para muchos el saldar un talonario es una forma de arte que raramente sale satisfactoriamente, para otros es una ciencia matemática donde cada diferencia de un austral se merece una búsqueda de, como mínimo, una hora. En cualquier caso, una computadora hogareña con un programa de balance de cuentas puede hacer el proceso más rápido y sencillo.

## Administración de las inversiones:

Hoy en día, las familias pueden tener dinero en diferentes fondos de la jubilación debido a que esos fondos no han sido transferidos cuando uno de sus integrantes cambia de empleo.
También pueden existir una o más cuentas del fondo particular de jubilación, asi como inversiones desde el mercado monetario hasta la participación en una determinada empresa. Alguna de estas inversiones le dará anualmente un cierto monto de dinero.
Otras, en cambio, arrojarán montos que dependen de variables de ajuste y aquellas que le dará poco o ningún rendimiento anual.
La computadora hogareña puede estar al tanto de sus inversiones, ingresos actuales y de los ingresos probables para el futuro. Si usted toma sus propias decisiones sobre en qué invertir, existen varios programas que lo ayudan en el proceso de la evaluación. Algunos programas, por ejemplo, evalúan la compra y venta de acciones, otros comparan rendimiento neto (impuestos deducidos) de diferentes tipos de inversiones.

## Inventario del hogar:

Usted, seguramente, no está esperando un desastre de la naturaleza, o fuego que destruya su casa. Por supuesto tampoco espera ser robado. Pero si esto ocurriese, sería capaz de proveer a su compañía de seguros toda la información correcta para poder cobrar el seguro? Un programa en una computadora hogareña puede llevar correctamente sus propiedades actuales.
Este puede estar al tanto de la localización de sus bienes.
Usted podría rápidamente saber dónde se encuentra un determinado objeto como ser un cassette, disco, regalos, etc.

## Gestión de base de datos:

Los programas de tratamiento de bases de datos están asociados más bien con computadoras de gestión que con computadoras hogareñas. De todas maneras hay varios usos para este tipo de programas. Supongamos que usted lleva una agenda o archivo sobre sus familiares, amigos y conocidos. En un determinado momento, desea saber la dirección de un familiar que hace bastante tiempo que no lo ve. Debe pues empezar a buscar su dirección en la agenda rogando que allí esté. De otra forma tendrá que empezar a buscar en toda la correspondencia alguna carta que él le haya enviado para saber cuál es su dirección actual. El programa de gestión de base de datos administra todo tipo de información como la que acabamos de describir. Además usted puede especificar las características de las personas que constituyen su base de datos para luego interrogar por todas las que verifiquen determinados rasgos.
Además, un simple programa de "mailing list" (envío de cartas "masivo") le permitirá especificar si una dirección pertenece a un amigo a un familiar.

## Decisiones financieras:

Imagine por un momento la posibilidad de comprar un coche en cuotas. Usted puede obtener el préstamo para la compra con intereses muy bajos o puede percibir un descuento por compra al contado y organizar, de esta manera, su propia financiación. Qué situación sería, para usted, la mejor? Varios programas utilitarios le ayudarán a analizar el problema y tomar la mejor opción. También usted puede preguntar "que ocurriría si....".
Rápidamente el programa responderá su pregunta informándole el dinero que usted tendría que invertir para llevar a cabo esa posibilidad.

## Software del Drean Commodore 64 para las finanzas y contabilidad domésticas

El software disponible para la finanzas y



contabilidades domésticas no es abundante, pero existen programas muy buenos.

Nosotros comentaremos algunos de los programas disponibles para la Drean Commodore 64. Por la facilidad de operación de algunos paquetes de software, los fabricantes de éstos decidieron poner la palabra "easy" (fácil) delante del nombre específico. Es así como se encuentran el Easy Finance I,II,III,IV y V. El primero de ellos lo ayuda en análisis de costo de préstamos.

El segundo le ayudará a tomar decisiones sobre sus inversiones.

Por ejemplo usted desea ahorrar e invertir dinero durante un determinado tiempo. De acuerdo al dinero que usted espera al finalizar ese período, el programa le dirá lo que debe invertir para llegar a esa suma.

El tercer programa utilitario es una continuación del segundo.

Este le ayudará, además, a tomar decisiones sobre acciones y bonos.

El cuarto y el quinto están orientados a un uso más profundo y no tanto a las aplicaciones domésticas.

Otro de los utilitarios disponibles para este tipo de aplicaciones es el "Presupuesto Familiar" desarrollado para la Drean Commodore 64. Su objetivo esencial es llevar un control sobre el presupuesto y el gasto efectuado en el hogar, aunque puede ser utilizado en otras entidades económicas, durante un horizonte de dos a doce períodos en base a movimientos de ingresos y gastos efectuados durante los meses. El programa permite hasta doce conceptos distintos, muestra cada uno de los ingresos y gastos en su total, además da los saldos mensuales y saldos acumulados. Se muestra también en forma de histograma los ingresos, gastos mensuales, y los saldos acumulados.
Al ejecutar este programa aparece, primeramente, un "menú" conteniendo cada una de las operaciones. Estas son Mantenimiento, Consulta, Archivo, Histograma, Fin.
El primer ítem efectúa un mantenimiento al archivo que se encuentra en la memoria del C-64. Es decir que se da de alta un ingreso o gasto, también aquí se da de baja o se modifica un dato si así se desea.

En el ítem de Consulta se muestra la tabla de presupuesto conteniendo: ingresos, gastos totales y saldos acumulados. La consulta es trimestral, así que aparecerán los tres en la pantalla a la vez. Podemos también consultar los meses anteriores o posteriores.
El tercer ítem (Archivo de datos) permite realizar operaciones como cargar un archivo, guardar un archivo, mostrar el directorio (sólo si se dispone de diskette), estado del disco, volver al menú principal.
En Histograma se mostrará en forma gráfica los ingresos y gastos totales así como del saldo acumulado en cada mes dentro del período analizado, dando una mayor idea sobre la economía familiar.
El último ítem (Fin) permite abandonar el sistema. De todas maneras se imprimirá un mensaje preguntándole si realmente desea salir (en caso de haber tipeado por error esa opción).
De todas formas acompaña a este utilitario un manual explicando con lujo de detalle cada una de las posibles operaciones de presupuesto Familiar.

J.D. Willis y M. Miller
"Guía práctica para el conocimiento del Commodore 64", ediciones Aura.

# Impresoras para la C-64

*En esta nota comentaremos las diferentes impresoras que hay en nuestro mercado para la Drean Commodore 64, informándoles las ventajas y desventajas de cada una de ellas, precios, etc. Queremos, de esta forma, que el usuario de la C-64 disponga de una serie de elementos de juicio a la hora de elegir una impresora.*

## COMPUPRINT RX80

Para utilizar esta impresora con la Drean Commodore 64 se requiere una interface paralela. Las siglas que representan su modelo determinan la compatibilidad con otras del tipo RX80. Es una de las pocas impresoras de grandes cualidades a bajo costo (menor que sus similares). Su principal característica es la velocidad de impresión de 120 cps bidireccional.
El sistema de arrastre del papel se realiza por fricción o por tracción. Este es de 80 columnas. Se puede comprimir el texto para que imprima un máximo de 136 columnas. Además dispone de un detector de falta de papel que pone automáticamente la impresora off line (fuera de línea) para que ésta no pueda recibir datos de la computadora, evitándose así el desgaste innecesario del cabezal y de la cinta. Esta última es de tipo cartucho, pudiéndose conseguir repuesto de él o, simplemente, recambiar la cinta.
Tiene modo gráfico, juntamente con diversos set de caracteres. Además de tener el set ASCII standar tiene siete set de caracteres internacionales. Estos son:

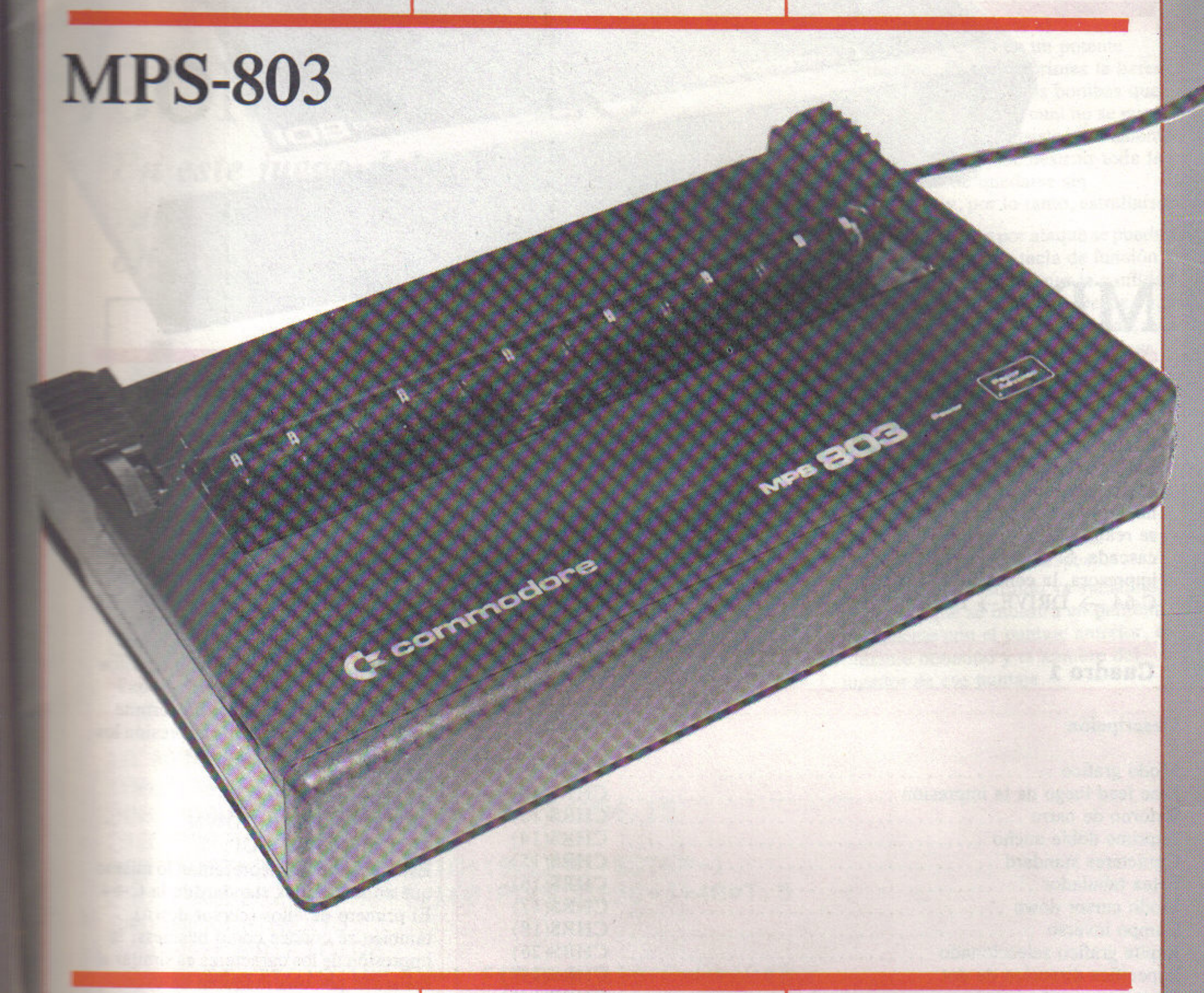
ASCII U.K (ASCII Inglaterra)
FRANCES
ALEMAN
SUECO
DANES
ITALIANO
ESPAÑOL

Cada uno de estos nuevos set disponen de todos los caracteres de la lengua respectiva. Por ejemplo el set ASCII ESPAÑOL tiene la "ñ" internamente. Para seleccionar el set correspondiente debemos setear una serie de llaves que se encuentran dentro de la impresora.

Por default la impresora asume que se trabajará con ASCII estándar. Dispone de caracteres especiales y científicos como ser doble ancho, enfatizado, remarcado, comprimido, italics, doble golpe, suscripto, sobre-escrito, mono unidireccional.
También dispone de caracteres de comandos orientados a la impresión en sí como ser avanzar una línea, saltear una línea, avanzar hasta la próxima hoja, cambiar localización del inicio de la próxima hoja, definir líneas por hoja, definir caracteres por pulgada, definir tabulador horizontal, tabulador vertical, marcar los renglones horizontales, marcar renglones verticales.
Tiene comandos orientados al manejo de gráficos tales como imprimir en simple densidad, imprimir gráficas en densidad doble, imprimir gráficas en densidad doble con velocidad doble, elección de 60 o 240 puntos por pulgada. Existen otros tipos de comandos orientados al funcionamiento de la impresora, los cuales activan y desactivan el indicador de fin de papel, indicador on/off line entre otros.
Sus características técnicas son:

| | |
|---|---|
| SISTEMA DE IMPRESION | IMPACTO POR MATRIZ DE PUNTOS |
| INTERFACE | PARALELO (COMPATIBLE CENTRONICS) ESTANDAR<br>RS 232 (OPCIONAL) |
| CARACTER | MATRIZ DE 9x9 PUNTOS |
| VELOCIDAD DE IMPRESION | 120 CPS BIDIRECCIONAL |
| TIPOS DE CARACTERES | 96 CARACTERES ASCII<br>88 CARACTERES INTERNACIONALES<br>96 CARACTERES ITALICOS<br>64 CARACTERES ESPECIALES<br>32 CARACTERES DE BLOCK<br>96 CARACTERES PROGRAMABLES |

# MPS-803



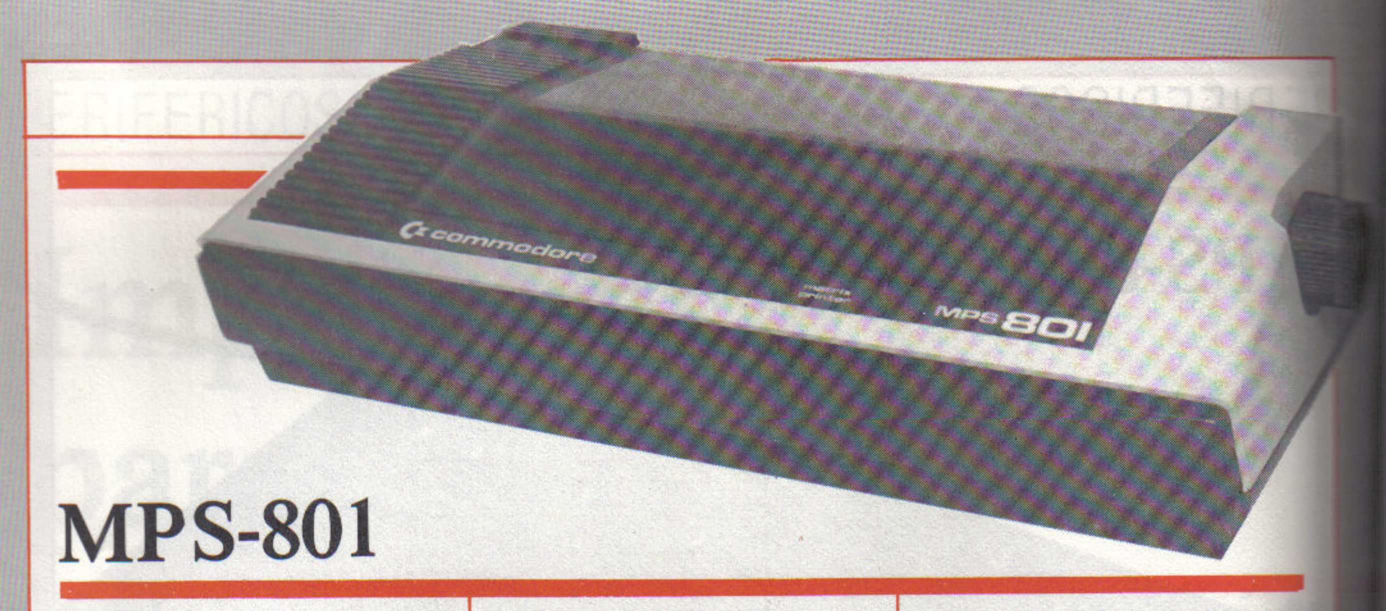

La 803, al igual que la 801, no necesita interface externa ya que es totalmente compatible. Las diferencias más importantes que tiene con aquella es que su velocidad de impresión es de 60 caracteres por segundo y, fundamentalmente, que es bidireccional. El sistema de movimiento del papel es a fricción siendo opcionalmente posible movimiento a tracción. Dispone de modo gráfico y contiene el mismo set de caracteres que la 801.

Los códigos para determinar un tipo de impresión también son iguales a la última citada. Cada caracter está formado por una matriz de 7x6.

Dispone de una llave selectora la cual determina el canal lógico del archivo por el cual estará asociado a la C-64. Este puede ser 4 o 5. Dispone de otra llave la cual selecciona la separación entre línea y línea. Otra de las diferencias que presenta con su similar es el cartucho de cinta el cual es un poco más grande conteniendo en su interior más cinta implicando, así, mayor vida útil.

## Características Importantes.

| | |
|---|---|
| VELOCIDAD DE IMPRESION ... | 60 CPS |
| BIDIRECCIONAL | |
| CANTIDAD DE COLUMNAS ... | 80 |
| VIDA UTIL DE LA CABEZA DE IMPRESION ..................... | 20 MILLONES DE CARACTERES |
| PESO .......................... | 2 Kg |
| DIMENSIONES ................. | 330x190x70 |
| ALIMENTACION ................ | 110V/220-240V 50-60Hz |
| CONSUMO ..................... | 30 W DURANTE IMPRESION<br>8 W EN REPOSO |
| RANGO DE TEMPERATURA ... | 5 A 35 C |




# MPS-801

No se requiere interface externa para conectar esta impresora ya que es totalmente compatible con la C-64. Esta conexión se realiza a través de la PORT serie en el DISK DRIVE. Toda nueva inclusión de dispositivos externos se realiza, en la C-64, en forma de cascada. Si se dispone de drive e impresora, la conexión se realiza: C-64 —> DRIVE—> IMPRESORA—> —>OTROS PERIFERICOS

Trabaja con papel de ochenta columnas. Como el movimiento de éste es a tracción, se requiere de formularios continuos especiales que pueden ser hallados en nuestro mercado. Su precio oscila en los 9 australes la resma de 1000 hojas. El sistema de cinta es a cartucho formado por una cinta sin fin y el tanque de tinta. Actualmente existe respuesta de ésta. Más aún, es posible cambiar todo el cartucho o solamente la cinta. Los precios respectivos andan alrededor de los 8 y 4 australes respectivamente. Este es uno de los detalles más importantes cuando se debe comprar una impresora; saber si hay repuesto para la cinta. La velocidad de impresión es de 50 caracteres por segundo (unidireccional). La impresora dispone de modo gráfico. También podemos fácilmente definir nuestros propios caracteres. Cada uno de éstos está formado por una matriz de 6x7 denominados dot matrix. Se permite solamente cuatro tipos de impresión los cuales son:

DOBLE ANCHO
CAMPO INVERSO
MODO CURSOR DOWN
MODO CURSOR UP

Estos dos últimos representan lo mismo que en los modos standardde la C-64. El primero de ellos (cursor down) también se conoce como business; la impresión de los caracteres es similar al realizado por una máquina de escribir. El modo de campo inverso es similar al CNTRL 9 y CNTRL 0 (RVS ON-OFF) de la C-64. Finalmente el doble ancho imprime los caracteres ampliados (largo y ancho). Cada uno de estos modos se transmiten a la impresora a través del comando CHR$ seguida por el código respectivo. Estos son según Cuadro 1

Ellos se envían a través del número de archivo lógico 4. Se puede cambiar este número a 5 previamente conmutando una llave externa que tiene la impresora para tal fin. Otra de las posiciones de esta llave es TEST la cual origina un auto-chequeo de la impresora. Tiene también un interruptor soft-tach el cual avanza la hoja automáticamente. Esto se puede realizar también manualmente a través de la rotación de un rodillo diseñado a tal efecto. Las especificaciones técnicas de la impresora son según cuadro 2

**Cuadro 1**

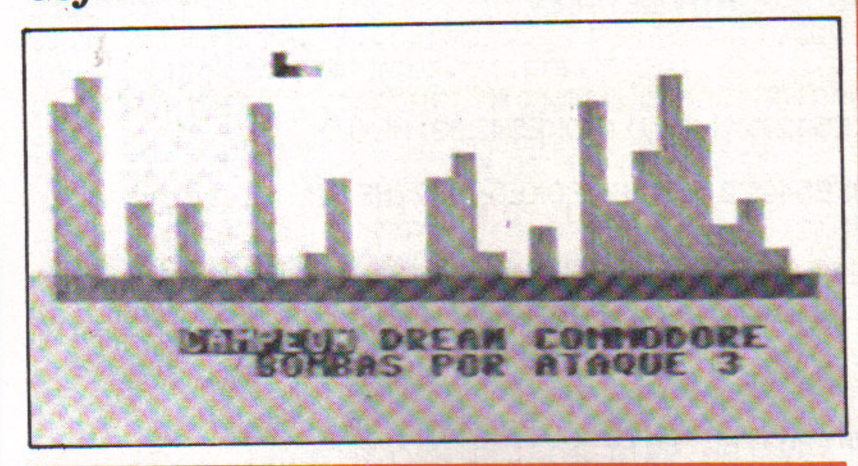
| Descripción | Código |
|---|---|
| Modo gráfico | CHR$(8) |
| Line feed luego de la impresión | CHR$(10) |
| Retorno de carro | CHR$(13) |
| Imprime doble ancho | CHR$(14) |
| Caracteres standard | CHR$(15) |
| Setea tabulador | CHR$(16) |
| Modo cursor down | CHR$(17) |
| Campo inverso | CHR$(18) |
| Repite gráfico seleccionado | CHR$(26) |
| Especifica dirección de dot | CHR$(145) |
| Desactiva campo inverso | CHR$(146) |

**Cuadro 2**

| | |
|---|---|
| METODO DE IMPRESION | IMPACTO DE MATRIZ |
| CARACTER | MATRIZ DE 6x7 |
| CARACTERES | NUMERICOS, SIMBOLOS, CARACTERES GRAFICOS, MODOS UPPER/LOW |
| CODIGO DE CARACTERES | ASCIICBM |
| VELOCIDAD DE IMPRESION | 50 CARAC./SEG. UNIDIRECCIONAL |
| MAX NUMERO DE COLUMNAS | 80 |
| DIMENSIONES | 237x438x115 |
| PESO | 4,8 Kg |
| REQUERIMIENTOS DE TENSION | 120 V |
| CONSUMO | 25 W EN IMPRESION 8 W EN REPOSO |
| TEMPERATURA DE OPERACION y | 5 A 40 C |

# BOMBARDEO

***En este juego deberemos comandar valientemente un avión que tiene por objetivo destruir las bases enemigas.***




El programa convierte el teclado de la C-64 en un potente bombardeo. Cuando oprimas la barra espaciadora, soltarás las bombas que caerán en la ciudad la cual no se puede defender. El juego consiste básicamente en un avión que debe destruir toda la ciudad antes de quedarse sin combustible y, por lo tanto, estrellarse.

El total de bombas por ataque se pueden variar oprimiendo la tecla de función 1 (F1). Podemos seleccionar la cantidad de bombas que querramos. Desde ya, a medida que tengamos más bombas, más grande será la ciudad a bombardear implicando que el ataque será más difícil.

El juego almacena los diez puntajes más altos y los imprime a medida que culmina un juego. Las bombas están representadas por flechas y se requiere del jugador buenos reflejos para lanzarlas justo en el momento adecuado, ni antes ni después.

Cada edificio destruído va sumando puntos lo que se muestra en pantalla juntamente con el puntaje anterior, el máximo obtenido y el nombre del jugador de ese puntaje.

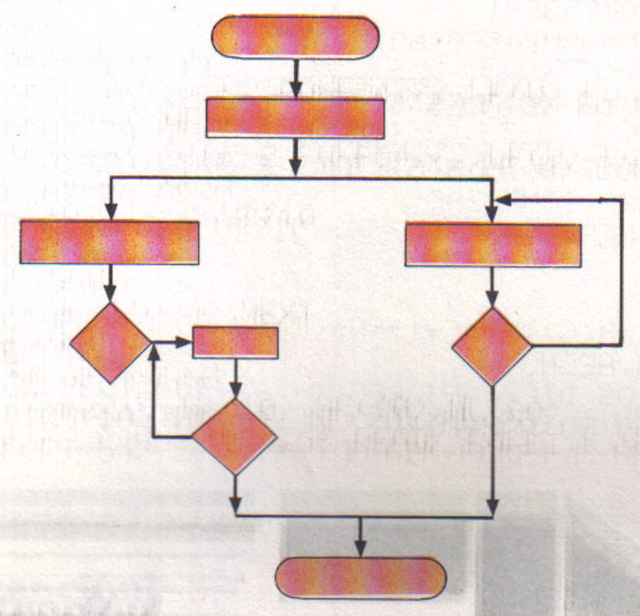
```
10 REM *** BOMBARDEO ***
100 GOSUB1000
200 GOSUB2000:GOSUB710:POKE54296,15
210 FORJ=1TO30
220 FORT=20TORND(1)*10+(9-LEV)+5STEP-1
230 POKE1024+J+40*T,160:POKE55296+J+40*T,5
240 NEXTT:NEXTJ
249 REM EDIFICIOS
250 FORB=7TO19:H=LEV+1:FORA=0TO39:O=A+40*B
260 GETA$
270 IFA$=" "THENGOSUB500
275 IFPEEK(1025+O)=160THEN4000
280 POKE1023+O,32:POKE55295+O,6
290 POKE1024+O,252:POKE55296+O,6
300 POKE1025+O,98:POKE55297+O,6
320 NEXTA:GOSUB400:NEXTB
330 GOSUB800:GOTO210
399 REM AVION
400 POKE1023+O,32:POKE55296,2
440 RETURN
499 REM DISPARO
500 H=H-1:IFH<1THENRETURN
505 FORQ=B+2TO19:W=A+40*Q
510 IFPEEK(1024+W)=160THENGOSUB600
520 POKE1024+W-40,32:POKE55256+W,2
530 POKE1024+W,30:POKE44296+W,2
540 NEXTQ
550 RETURN
599 REM SUMA DE PUNTAJE
```

```
600 S=S+10
610 PRINT"▒ PUNTOS";S
620 POKE54276,129
630 POKE54272,HF(Q-7):POKE54273,HL(Q-7)
640 FORT=1TO1:NEXT:POKE54276,0
650 RETURN
669 REM AVISO
710 PRINT"▒ PUNTOS";S;TAB(12)"  MAXIMO";H(1)
720 PRINT"▒▒ ";TAB(14)"ULTIMA PUNTUACION";L
730 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";TAB(6);"▒CAMPEON▒ ";H$(1
735 PRINTTAB(8)" BOMBAS POR ATAQUE";LEV
740 RETURN
799 REM
800 POKE54276,17
805 FORU=3TO7STEP2:POKE54272,LF(U):POKE54273,HF(U)
807 GOSUB870:NEXT
810 FORU=7TO3STEP-2:POKE54272,LF(U):POKE54272,HF(U)
813 GOSUB870:NEXT
820 PRINT"▒":GOSUB710
830 POKE54276,0
860 RETURN
870 FORUP=1TO200:NEXT
880 RETURN
999 REM VECTORES
1000 DIMH(10),H$(10),HF(20),HL(20)
1010 FORT=54272TO54285:READK:POKET,K:NEXT
```

```
1020 FORT=1TO20:READHF(T),HL(T):NEXT
1030 FORK=1TO10:H$(K)="DREAN COMMODORE"
1040 NEXT
1090 RETURN
1999 REM
2000 L=S:S=0
2080 POKE53280,7:POKE53281,7:PRINT"■"
2085 GOSUB6000:PRINT"■"
2090 RETURN
3999 REM
4000 POKE1024+O,102:POKE55296+O,2:POKE54276,129
4010 POKE43272,75:POKE54273,34:FORYU=1TO500:NEXT:POKE4276,0
4040 PRINT"■":POKE53280,4:POKE53281,7
4050 FORI=1TO10:IFS>H(I)THENGOSUB4500:GOTO4070
4060 NEXT
4070 PRINT"■";TAB(12)"■■LOS 10 MEJORES"
4080 PRINT"■■"
4090 FORO=1TO10:PRINTTAB(5)O;TAB(10)H(O);TAB(20)H$(O):NEXT
4100 PRINT"■■■ OPRIMA F1 PARA COMENZAR"
4110 GETR$:IFR$<>CHR$(133)THEN4110
4120 GOTO200
4499 REM
4500 FORC=9TOISTEP-1:H(C+1)=H(C):H$(C+1)=H$(C)
4510 PRINTTAB(10)"■■■■■BIEN HECHO"
4520 PRINT"■■■■ESTA ENTRE LOS DIEZ MEJORES"
4530 PRINT"■■■■■■■■■■ESCRIBA SU NOMBRE"
4540 INPUTH$(I)
4543 H$(I)=LEFT$(H$(I),19)
4560 H(I)=S:I=10
4570 RETURN
5999 REM FIN BLOQUE1
6000 REM INICIO BLOQUE2
6010 PRINT"■":F$="BOMBARDEO":LEV=1
6020 FORUI=1TO37:PRINTTAB(UI)"■ ■■■"
6030 IFUI>14THENPRINTTAB(15)"■";LEFT$(F$,UI-14)
6035 FORGH=1TO10
6040 NEXTGH:NEXTUI:PRINT"■■";TAB(38)"   "
6045 PRINTTAB(7)
6050 PRINT"■■■■■■■■■■■■";TAB(12)"■BOMBAS POR ATAQUE";LEV
6060 PRINT"■■■■■■■■■■■■■  DESTRUYA LOS EDIFICIOS"
6070 PRINT"■■■■■■    DISPONE DE POCAS BOMBAS"
6080 PRINT"■■■LA BARRA ESPACIADORA SUELTA LAS BOMBAS"
6090 PRINT"■■■■■■  F1 CAMBIA EL NUMERO DE BOMBAS"
6095 REM
6110 PRINT "■■■OPRIMA LA BARRA ESPACIADORA PARA EMPEZAR"
6120 GETD$
6130 IFD$=" "THENRETURN
6140 IFD$<>CHR$(133)THEN6120
6150 LEV=LEV+1
6160 PRINT"■■■■■■■■■■■■";TAB(12)"BOMBAS POR ATAQUE";LEV:GOTO6120
6165 REM DATAS
10000 DATA0,0,0,0,129,255,255,0,0,255,15,65,255,255
10010 DATA7,53,8,23,8,147,9,159,10,205,11,114,12,216,14,107,16,
      47,17,37,19,63
10020 DATA21,154,22,227,25,177,28,214,32,94,34,75,38,126,43,52,
      45,198
```

# MAPA DE MEMORIA

***Continuamos describiendo las direcciones de memoria de la Drean Commodore 64.***



## DIRECCIONES $0017-$0018 (23-24)

Estas direcciones se utilizan como puntero a la dirección del último string en el stac temporario de strings.

## DIRECCIONES $0019 - $0021 (25-33)

Las direcciones de memoria comprendidas dentro de este rango se utilizan para describir la información correspondiente a strings temporales; es decir aquellos que no se asignan a variables string. Un ejemplo de ello es el literal "ABCD" correspondiente al comando PRINT"ABCD". La información antes nombrada corresponde a la longitud, dirección de comienzo y de fin del string en cuestión.

## DIRECCIONES $0022 - $0025 (34-37)

Esta área es utilizada por algunas rutinas del BASIC. Aquí se almacenan temporalmente punteros y resultados de operaciones que ellas realizan.

## DIRECCIONES $0026 - $002A (38-42)

Estas direcciones son utilizadas por las rutinas de división y multiplicación. También son utilizadas por la rutina que calcula la longitud del área de memoria a reservar cuando se ejecuta una sentencia DIM.

## DIRECCIONES $002B-002C (43-44)

Este puntero le permite al BASIC conocer en dónde comienza el programa basic. Comúnmente éstas apuntan a la dirección $0801 (2049). Es decir que el contenido de la dirección $002B contendrá $01(byte bajo) mientras que el de la $002C será $08 (byte alto). A través de este puntero es posible modificar la dirección de comienzo de nuestro programa. De esta manera podríamos, por ejemplo, compatibilizar la distribución de memoria del C-64 con otras computadoras COMMODORE en caso de interconectarlas o de efectuar otro tipo de tarea, ejecutar dos o más programas al mismo tiempo. Esto se logra commutando sucesivamente este puntero a los comienzos de cada uno de los programas involucrados (se requieren de más detalles para llevar a cabo esto último). Finalmente este puntero nos permite grabar en diskett o cassette una sección de nuestro programa. Esto se debe a que la rutina SAVE mira en estos dos bytes el comienzo del Programa basic que será grabado en el periférico actual.

## DIRECCIONES $002D-$002E (45-46)

Estas direcciones apuntan a la dirección donde culmina el programa basic. Las avariables que utiliza nuestro programa se almacenan a continuación del texto basic. Esto implica que estas direcciones apuntan al comienzo de la zona antes mencionada. Toda variable que no represente arreglos se almacena aquí. Cada una de estas variables utilizan 7 bytes. Los dos primeros corresponden al nombre de la misma representada en código ASCII. Esto nos dice que los caracteres significativos de una variable son los dos primeros. En caso de utilizar una variable cuyo nombre esté formado por un solo caracter, el segundo byte será 0 Seteando el séptimo bit de uno o ambos bytes, se determina el tipo de variable. Si en ninguno de los dos se seteó el bit 7, significa que se trata de una variable de coma flotante o real. Si, en cambio, el bit séptimo del primer byt fue seteado (puesto a "1", sumado 128), indica que se trata de una variable tipo string. Si el segundo byte tiene seteado el séptimo bit, la variable corresponde a una función definida (DEF FN).
Finalmente si en ambos bytes se encuentra seteado el bit 7, significa que la variable en cuestión es tipo entera.
Los restantes cinco bytes dependerán del tipo de variable. Si ésta es de punto flotante, el número se representará acorde a ello (singo, mantisa, exponente, etc.). Lo mismo ocurre con variables enteras. Si la variable es string, esos cinco bytes representan lo siguiente:
El tercer byte indica la longitud de la cadena. El cuarto y quinto byte apuntan a la dirección donde se almacena esa cadena. El sexto y séptimo aquí no se utilizan.
Si la variable representa a una función definida, el tercer y cuarto byte apuntan a la dirección en el texto basic donde

comienza la función definida. El quinto y sexto byte apuntan a la dirección donde se encuentra la variable independiente (caso del X en FN A (X)). El último byte no se utiliza.
Las variables son almacenadas en orden de aparición; es decir a medida que son creadas en el programa basic. Cuando el intérprete busca alguna de ellas, realiza una busqueda secuencial a partir del comienzo de esta área. Otro detalle importante para resaltar es que las variables que representan arrays (vectores, matrices, etc.) se almacenan a partir del final de la anterior área. Es decir que si en un programa trabajamos con variables comunes (no arrays) y variables arrays y si, en el medio del programa, creamos una nueva variable (no array), el sistema operativo debe mover siete byte hacia arriba para hacer lugar a la nueva variable creada. Es por ello conveniente definir todas las variables no arrays antes de las array.
Un dato importante es que la dirección de almacenamiento del actual texto string que esta siendo utilizado por el programa se encuentra en las direcciones $0033 y $0034 (punteros).

## DIRECCIONES $002F-$0030

Estas direcciones apuntan a la dirección de fin de la variable no arrays, donde comienza la zona de almacenamiento de la variables arrays. El formato utilizado para almacenar estas variables es el siguiente:
Los primeros dos bytes representan el nombre de la variable (de igual forma que en el caso de variables no arrays, excepto que aquí no hay equivalencias para el caso de funciones definidas)
Luego sigue un offset de dos bytes utilizado para indicar el comienzo del próximo array (primero el byte bajo).
Luego hay un valor de un byte el cual representa el número de arrays dimensionados (por ejemplo un 2 indica que se trata de un arreglo bidimensional es decir A(X,Y)). A continuación de éste se encuentran un par de bytes los cuales poseen el valor del dimensioamiento + 1 de cada array (de esta manera se puede almacenar el elemento 0 de un array como ser A(0). Finalmente se encuentran los valores de la variables. El formato utilizado para representar esos valores es el mismo que se utiliza para variables no arrays con la salvedad que cada valor toma el espacio requerido; es decir que variables de coma flotante toman cinco bytes cada una, las enteras dos bytes y los descriptores de string tres bytes.

## DIRECCIONES $0031-$0032 (49-50)

Estas direcciones apuntan a la dirección de fin del área de almacenamiento de arrays y, implícitamente, a la dirección de comienzo de la memoria (RAM) libre. Los caracteres que constituyen las variables strings se almacenan a partir del tope de memoria libre y van ocupando las posiciones consecutivas, hacia abajo, hasta llegar a la dirección que apuntan éstas.

## DIRECCIONES $0033 – $0034 (51-52)

Estas direcciones apuntan al actual dirección de fin del texto string (es decir los caracteres que constituyen a éstos) y al tope de RAM libre (como dijimos anteriormente, el texto string se almacena a partir del tope de memoria hacia posiciones de memoria menores). Todo texto string que se agregue en el programa basic se colocará en el área a la cual estas direcciones ($0033-$0034) apuntan. Las rutinas de reset,. ya sea el reset en sí o cuando prendemos la máquina, setean los contenidos de éstas para que apunten al tope de memoria RAM. El comando CLR los setea a la dirección final de memoria BASIC.

## DIRECCIONES $0035-$0036 (53-54)

Estas direcciones se utilizan como punteros temporales a la dirección de comienzo del último string que se sumó o que se desplazó en memoria.

## DIRECCIONES $0037-$0038 (55-56)

Estas direcciones apuntan a la dirección más alta utilizada por el BASIC. Este puntero lo podemos utilizar para definir áreas dentro de la memoria libre que no sean perturbadas por el BASIC. Por ejemplo si deseamos disponer de un área de 1 kb en el tope del BASIC debemos poner en la primera línea de nuestro programa:

```
POKE 56, PEEK (56) - 4: CLR
```

El CLR es necesario para asegurar esa área de memoria libre.

## DIRECCIONES $0039-$003A (57-58)

Estas direcciones contienen el número de línea de la sentencia basic que actualmente se está ejecutando. En la dirección $0039 está el byte bajo mientras que en $003A se encuentra el byte bajo. Es decir que si está ejecutando la línea 10, los contenidos de estas direcciones serán:
($0039)=$00
($003A)=$0A ; $0A=10
Un valor de $FF en la dirección $003A significa que el BASIC se encuentra en modo directo. Las sentencias BASIC que no pueden ejecutarse en modo directo chequean esta dirección para determinar el modo actual. En caso de encontrarse con $FF se imprimirá el correspondiente mensaje de error.
Cuando se está en modo RUN, esta dirección contiene el número de línea del programa basic que actualmente se está ejecutando. Gracias a esta dirección podemos agregar el comando TRACE. Para ello debemos modificar el vector de la dirección $0308. En futuras notas explicaremos con más detalle cómo ampliar las funciones del basic residente.
El número de línea es utilizado por los mensajes de error y por BREAK. El valor que aquí se encuentra (el número de línea actual) es copiado en la dirección $003B por las rutinas de STOP, END, CONT y la tecla de STOP-BREAK.

## DIRECCIONES $003B-$003C (59-60)

Aquí se almacena la última línea del programa ejecutada antes de la sentencia END y se restablece la dirección $0039 por la rutina CONT.

## DIRECCIONES $003D - $003E (61-62)

Estas direcciones contienen la dirección de la sentencia basic que está siendo ejecutada. El contenido de la dirección $007A (apunta a la dirección del próximo carácter del texto basic) se almacena en esta dirección ($003D-$003E) cada vez que una nueva línea está siendo ejecutada.

## DIRECCIONES $003F - $0040 (63-64)

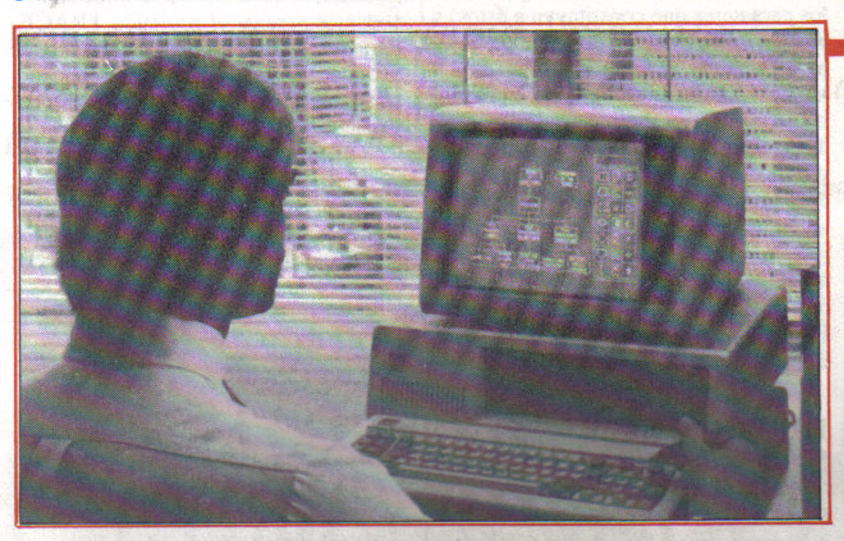
Esta dirección contiene el número de línea actual de la sentencia DATA que está siendo leída por la sentencia READ De esta manera si ocurre un error concerniente al proceso READ-DATA, se podrá determinar qué sentencia DATA ocasionó el error. (en qué línea se encuentra)

## DIRECCIONES $0041 - $0042 (65-66)

Estas direcciones apuntan a la dirección dentro del texto basic donde una sentencia DATA está siendo "ejecutada" por una sentencia READ.

# INGRESO CONTROLADO DE DATOS

***Este software se utiliza como una subrutina que sólo permite ingresar información cuyas características verifiquen ciertas condiciones.***



Generalmente, al efectuar un programa debemos, en determinados puntos, realizar ingresos de datos desde el exterior; más precisamente desde la consola. Por algún motivo los datos a ingresar deben pertenecer a un determinado tipo: numéricos o alfanuméricos. Por eso tenemos que chequear que éstos pertenezcan al tipo preestablecido.

Esta tarea se realiza básicamente a través de las sentencias INPUT y GET. Esta última es ideal para los objetivos buscados, ya que nos permite controlar cada uno de los caracteres que se ingresan y así verificar su inclusión en el conjunto al cual ella pertenece. Agreguemos que la longitud de los datos está también preestablecida.

Para el caso de datos alfanuméricos simplemente utilizamos la sentencia LEN.

Para las numéricas el procedimiento no es tan directo; debemos efectuar algunos pasos previos. Por ejemplo, si tenemos que aceptar una fecha y asignarla a una variable numérica, el procedimiento para tomar solamente sus cuatro digitos sería:

```
10 INPUT FECHA
20 AUX=INT(FECHA/1000)
30 continúa el programa
```

El proceso de control de ingresos de datos demanda aún muchas más verificaciones antes de aceptarlos y asignarlos a variables.

El programa que aquí les ofrecemos se utiliza como una subrutina que sólo permite el ingreso de datos cuyas características verifiquen nuestras condiciones. Esas características se refieren al tipo de dato y su longitud. Como está escrito en lenguaje máquina su velocidad de ejecución es muy alta. Antes de llamarla, lo que se logra tipeando SYS 49152, debemos setear los parámetros que ella utiliza en determinadas direcciones de memoria y luego leer el dato ingresado desde el área asignada al buffer del cassette (aquí se utiliza a partir de la dirección 828). Los parámetros antes mencionados se colocan utilizando la sentencia POKE y las direcciones 253, 254. La primera indica la longitud máxima permitida (desde 1 hasta 191 caracteres) mientras que la segunda, el tipo de dato (0 numérica, 1 alfanumérica). Luego de ejecutarla ésta devuelve en el registro Y (dirección 782) la longitud del dato

Listado 1

```
1 REM ENTRADA CONTROLADA
5 FORX=0TO142
6 READA:POKE49152+X,A
7 CS=CS+A:NEXTX

8 IFCS<>21770THENPRINT
"ERROR EN LOS DATA.
CHEQUEE LOS VALORES.":STOP
9 PRINT"CONTROLADOR
ACTIVADO"
10 NEW
999 REM INSTRUCCIONES/
DATOS LM
1000 DATA 169,60,133,251,169,
3,133,252,160,0
1010 DATA169,166,145,251,200,
192,192,208,249,160
1020 DATA0,177,251,32,210,255,
169,157,32,210
1030 DATA255,152,72,169,0,32,
228,255,240,251
1040 DATA 170,104,168,138,201,13,
208,6,192,0
1050 DATA240,235,208,76,201,20,
208,22,192,0
1060 DATA240,255,136,32,210,
255,169,166,145,251
1070 DATA32,210,255,169,157,32,
210,255,208,207
1080 DATA196,253,240,203,72,
165,254,208,6,169
1090 DATA58,133,250,208,4,169,
91,133,250,104
1100 DATA24,201,48,144,2,176,10,
201,46,240
1110 DATA10,201,32,208,172,240,
4,197,250,176
1120 DATA166,145,251,200,32,210,
255,76,21,192
1130 DATA145,251,169,32,32,210,
255,169,13,32
1140 DATA210,255,96
```

ingresado. De esta manera podemos asignarlo a una variable.

El programa principal es el listado 1

Para ejemplificar el uso de esta rutina les explicaremos cómo trabaja el listado 2.

Listado 2

```
5 REM COMIENZO DEL
PROGRAMA BASIC
10 POKE253,15:REM SETEAMOS
LONGITUD
11 POKE254,1:REM SETEAMOS
TIPO
15 PRINT"[HOME][5 CR ABAJO]";
20 SYS49152:REM LLAMAMOS A
LA RUTINA
25 FORX=0TOPEEK(782)-1:REM
SACAMOS LOS CARACTERES
30 A$=A$+CHR$(PEEK(828+X))
35 NEXT
40 PRINT"[2 CR ABAJO]";
41 POKE253,5
42 POKE254,0
42 SYS49142
50 FORX=0TOPEEK(782)-1
55 B$=B$+CHR$(PEEK(828+X))
60 NEXT
65 PRINTA$;" ";B$
70 END
```

En la línea 10 se setea la longitud máxima de caracteres, en nuestro caso 15. En la línea 11 seteamos el tipo de dato (alfanumérico) poniendo el valor 1 en la dirección 254.

Luego, en la línea 20, llamamos a la rutina efectuando SYS 49152. Aquí aparecerá un caracter gráfico en lugar del cursor cuyo objetivo es indicarnos su disposición de aceptar los datos. Para este caso sólo podremos ingresar dígitos incluyendo el punto decimal.

Al oprimir RETURN la rutina devolverá el control al programa. Comenzaremos pues a leer los caracteres tipeados desde el teclado.

Utilizamos un lazo FOR-NEXT a través de una variable que va desde 0 hasta la cantidad de caracteres ingresados menos uno. Los tomamos a partir de la dirección 828 y los asignamos a la variable correspondiente (la variable X pudo haber ido desde 1 hasta el valor asignado a la dirección 782).

Para este caso se debe modificar de 828 a 827 en la línea 30.

Finalmente en las líneas 41 y 42 seteamos nuevamente las direcciones de memoria 253 y 254 para utilizar una vez más la rutina.

Nuevamente el tipo de dato a ingresar es numérico y la longitud máxima es cinco.

Todo intento de superar este valor máximo o ingresar algún caracter que no corresponda al tipo seleccionado, no será tenido en cuenta por la subrutina.

Otro detalle importante es que no se permiten ingresos de caracteres cuyos códigos ASCII superen a 90 o sean inferiores a 46, es decir que tampoco son tenidos en cuenta los caracteres especiales.

Para completar el concepto, ingresen el programa correspondiente al listado 2 y ejecútenlo (no olviden cargar y ejecutar antes la subrutina).

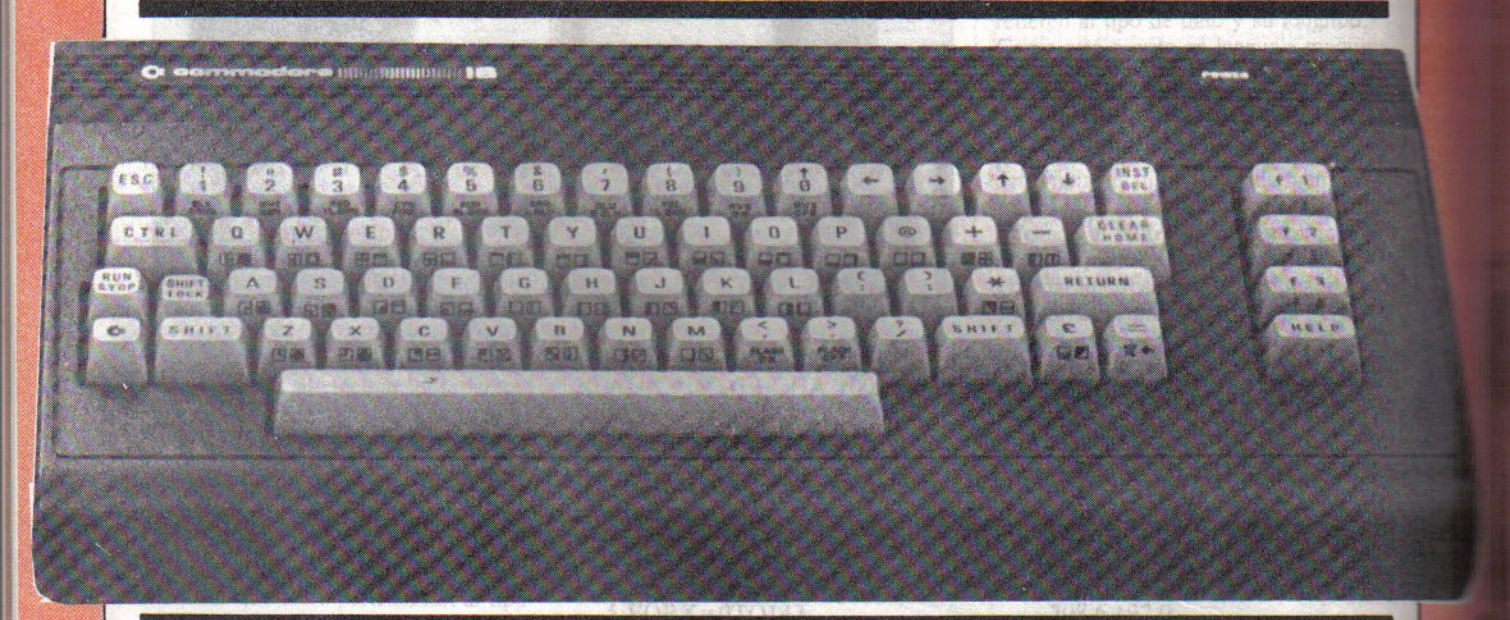
Hemos realizado algunas modificaciones en lo que respecta a caracteres especiales o de control del cursor. Por ejemplo, en la línea 15, se debe interpretar de la siguiente manera: abrir comillas, oprimir la tecla HOME y oprimir cinco veces la tecla que mueve el cursor hacia abajo. Finalmente deben cerrar comillas.

Consideramos que de esta manera los listados son mucho más fáciles de entender.

Esperamos poder aplicar este método en próximos números.

# DREAN COMMODORE 16: MAS QUE UNA HOME COMPUTER

***Les comentamos las características sobresalientes de este equipo, como su sistema "monitor", su manejo de gráficos, sonido, el novedoso teclado, el editor "full screen", y las notables posibilidades de expansión.***




No cabe duda que la Drean Commodore 16 es un equipo salido de las manos de la empresa Drean; basta con observar su aspecto externo. Excepto un discreto cambio de color, todo es similitud con el mueble del Drean Commodore 64.

También el nuevo y potente basic residente en el equipo es un sucesor de máquinas anteriores. Algo semejante ocurre con los periféricos: herencia de otros modelos en unos casos (compatible con el disk drive 1541) y modelos nuevos en otros (caso del nuevo datassette).

A pesar de que este computador fue lanzado para la enseñanza, sus grandes caracteristicas lo habilitan para realizar otras actividades como ser contables, científicas, etc. Otra de las características más sobresalientes que tiene este equipo es su sistema "monitor'6 el cual permite trabajar directamente en lenguaje máquina del microprocesador 7501. Los programadores de este lenguaje de bajo nivel, como lo es el lenguaje máquina, encontrarán en el "monitor" todas las herramientas necesarias para desarrollar y probar sus programas. Alguno de los comandos que dispone éste son el comando A el cual permite ingresar una instrucción en mnemotécnico del micro 7501 (similar al 6502), el comando T quien transfiere bloques de memoria específicos, etc.

Otra de las características que dispone el Drean Commodore 16 es su manejo de gráficos en alta resolución que, a diferencia del Drean Commodore 64, es de fácil manejo. El único inconveniente que surge como consecuencia de trabajar en este modo es que su memoria libre disminuye drásticamente a 2045 bytes. En notas anteriores hemos publicado los diferentes tipos de gráfico que se pueden obtener con la C-16.

Trabajando con su intérprete basic descubrimos lo versátil y potente que es no sólo para la programación sino, también, para depuración y manejo de errores. Su nuevo basic V3.5 desarrollado por Commodore, dispone de una gran cantidad de nuevos comandos. Podemos encontrarnos con estructuras del tipo WHILE donde se ejecuta una determinada tarea mientras se cumpla una condición; o el UNTIL que, a diferencia del anterior ejecutará un proceso hasta que una determinada condición sea satisfecha. También tiene el IF THEN ELSE estructura que no dispone la C-64 en su intérprete V2.0. Respecto a la depuración, la C-16

dispone del TRON/TROFF, comando que nos permite seguir línea a línea la ejecución de un determinado programa. El sonido se puede realizar fácilmente a través de los comandos SOUND y VOL. Este se emite a través del altavoz del receptor de T.V. para la obtención del sonido, el equipo dispone de dos canales: uno de sonido y otro seleccionable como generador de sonido o ruido. Para llevar a cabo esto último no es necesario acceder a determinadas áreas de memoria utilizando los comandos POKE y PEEK como ocurre en la C-64. Simplemente usamos los comandos antes descriptos. El comando SOUND permite seleccionar la voz, nota y duración mientras que VOL determina el volumen de ésta.
Si sumamos la posibilidad de poder definir "ventanas", comúnmente llamadas windows, a todas las características del equipo, nos encontraremos con un computador ideal para trabajar.

## Editor y Teclado

El C-16 comparte con los anteriores modelos de Drean Commodore un teclado de calidad poco frecuente en computadoras hogareñas de su precio, con teclas sólidas y distribuidas con la separación correcta para evitar el accionamiento simultáneo de más teclas de las deseadas. Por supuesto, también están presentes algunas teclas de función preprogramadas. Mediante ellas, es posible acceder directamente a algunas funciones y comandos, por lo general los más utilizados. El número de teclas de función es de cuatro, todas ellas dobles, que permiten acceder a ocho funciones preprogramadas. Debemos destacar una de ellas, denominada HELP. Tras la detención de la ejecución de un programa debido a un error, podemos rápidamente visualizar la clase a la cual pertenece con sólo presionar la tecla antes mencionada. Automáticamente se imprimirá el número de línea donde ocurrió.
El editor es el de los denominados "full screen". Permite editar cualquier línea del programa presente en pantalla. El método a seguir se reduce, sencillamente, a realizar las oportunas modificaciones utilizando las teclas de desplazamiento del cursor. Respecto a este último debemos resaltar que se dispone de cuatro teclas independientes para su movimiento.
A la hora de editar un programa disponemos de comandos como el RENUMBER o el AUTO, agregándose el comando DELETE quien nos permite borrar una serie de líneas de programa. Respecto al hardware del C-16, se dispone de un pulsador en el costado derecho de equipo cuya misión es la de resetear el computador sin necesidad de apagar la máquina.

## Capacidad de expansión

Las posibilidades de expansión no dejan de ser notables para un equipo que ha salido al mercado tan recientemente. Dispone de una unidad de cassette (modelo no estandar) permitiendo, además, la conexión de la unidad de disco 1541. Respecto a impresoras, el C-16 puede trabajar con cualquier impresora Commodore. Los joystick, o más exactamente los conectores de éstos, no son compatibles a los de la C-64. De todas maneras podemos fácilmente adquirir todo tipo de interfase para este equipo.
No hace falta decir que contar con una empresa como lo es DREAN S.A. respaldando la C-16 y C-64 es una gran garantía para todo actual y futuro usuario de los equipos Commodore. En rigor debemos también citar a la empresa PEEK que, gracias a su servicio técnico especializado desarrollando software y hardware para ambos equipos, posibilita el óptimo funcionamiento de las dos grandes Homes Computers.

## Especificaciones técnicas

| | |
|---|---|
| MICROPROCESADOR | 7501 |
| FRECUENCIA DE RELOJ | De 0.89 a 1.76 MHz |
| LENGUAJE RESIDENTE | BASIC 3.5 |
| MEMORIA RAM | 16 Kbytes |
| MEMORIA ROM | 32 Kbytes |
| TECLADO | 66 teclas en total.<br>Teclado ASCII en disposición QWERTY.<br>4 teclas para desplazamiento del cursor.<br>4 teclas dobles de función |
| EDITOR | Full screen (de pantalla) |
| RESOLUCION MAXIMA | Modo gráfico: 320x200<br>Modo texto: 25x40 caracteres |
| CARACTERES | Dos juegos: uno de 127, otro de 57 caracteres |
| COLOR | 121 colores |
| SONIDO | 2 canales: uno de sonido,<br>otro de sonido o ruido |
| SALIDAS | Expansión ROM<br>Conector para unidad de disco<br>Conexión para cassette VIC 1531<br>Toma para monitor<br>Toma para T.V.<br>E/S audio<br>Toma para alimentación |
| DIMENSIONES | 76.2x406.4x203.2 |
| ALIMENTACION | 220 v/50-60Hz |
| CONSUMO | 8 watts |

## Conclusiones

El DREAN COMMODORE 16 es un equipo que presenta interesantes novedades respecto a su predecesor, el VIC 20. Se ha prestado una especial atención a ciertos detalles antes olvidados. Entre ellos cabe destacar la gran cantidad de comandos BASIC (probablemente llega a duplicar el repertorio del VIC 20) y la flexibilidad de muchos de ellos, lo que contribuye a incrementar la potencia de su intérprete. Otros detalles importantes son la generación de sonido, gráficos especiales, estructuras de control y el "monitor" residente. También debemos incluir las teclas preprogramadas, cuya presencia facilita el trabajo, ya que es posible la introducción de ciertas órdenes con la sola acción sobre una tecla.
El único "pero" verdaderamente importante es la escasez de memoria RAM. Aunque al conectar el aparato la zona de memoria libre se eleva a 12277 bytes, la selección de los modos gráficos reduce el espacio disponible, situándolo a 2045 bytes; un volumen de RAM en la que parece imposible efectuar un programa de una determinada complejidad. Esperemos que en un futuro no muy lejano, DREAN COMMODORE subsane este inconveniente ampliando la memoria libre o desarrollando un periférico expansor.

# QUE PUEDO HACER CON UNA COMPUTADORA?

***Usos comerciales, profesionales y educativos que se le pueden dar a la Drean Commodore 64.***



Como ustedes saben —nos relataba Julio— mi señora y yo siempre le dimos a nuestros hijos todo lo que estaba a nuestro alcance. Cuando eran chicos les procuramos un triciclo a cada uno.

Después, cuando crecieron, les reparamos la bicicleta. Ahora, por curiosidad y entusiasmo, les compramos una computadora.

Nos metimos en el plan y la sacamos por sorteo en el primer mes. Todas nuestras expectativas estaban ahí, sobre una mesa. La prendimos. En la pantalla de nuestro televisor aparecieron unas palabras en inglés. Mi hijo menor oprimió algunas teclas mientras que mi señora y yo nos miramos sin comprender absolutamente nada. Mi hijo menor seguía oprimiendo teclas sin obtener un resultado positivo. Luego, mientras descansaba, me preguntaba: ¿y ahora qué?; ¿qué hago con una computadora?

Tal vez nuestro amigo, por sí solo y sin asesoramiento (o con un mal asesoramiento), comprendió que la única manera de sacarle provecho a la computadora era a través de un joystick; es decir jugando con los video games.

De esta forma él solo podía catalogar a la computadora como una buena jugadora; o sea, sólo le otorgaba adjetivos que denotaban astucia, ardid. Nosotros, a través de esta nota, queremos contestar en parte la pregunta original: todo lo que se puede hacer con un equipo Drean Commodore.

Para un pequeño comerciante, la Drean Commodore puede llevar exactamente el stock de la mercadería. Además puede repararla de tal manera que le informe cuáles fueron las ventas de cada día, es decir qué cantidad de mercadería se vendió (cuánto de este producto, cuánto de aquél, etc.), y cómo quedó el stock al finalizar la jornada para que pueda reponer los artículos faltantes.

Si desea conservar esta información para luego ver cual fue su rendimiento mensual, puede guardar todos estos datos en cassette, utilizando para ello el Datassette, o en disco a través del Drive 1541.

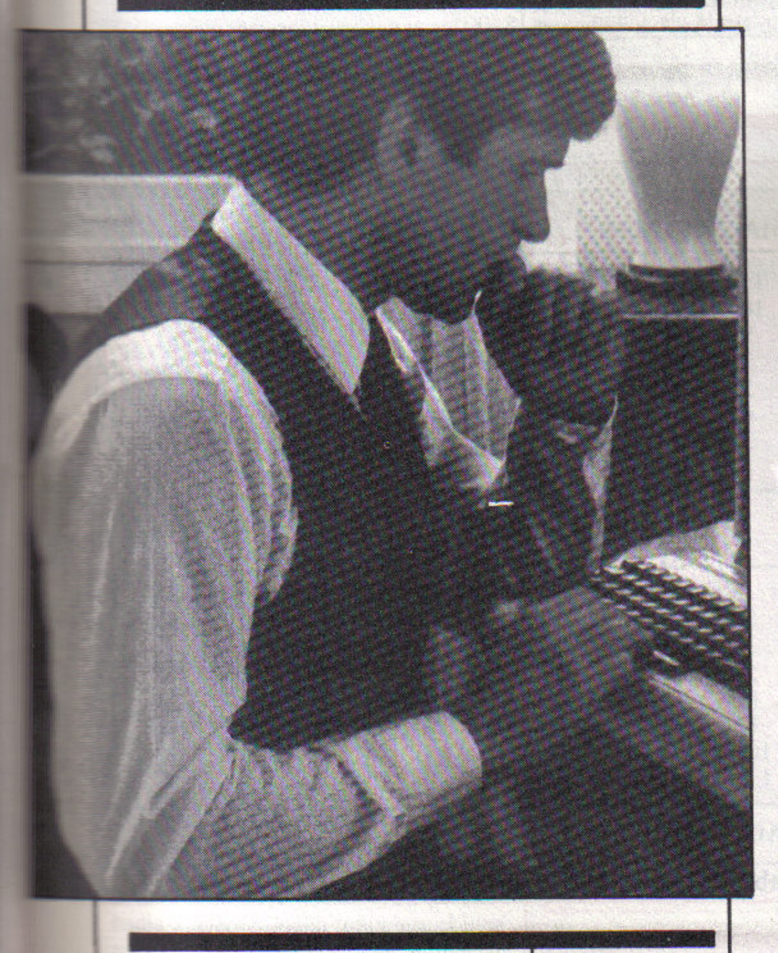
También le permite determinar rápidamente cuáles fueron sus ganancias en el día (incluyendo el pago a los empleados, gastos extras, etc.).

Supongamos que ese pequeño comerciante tiene un autoservice. Podría utilizar la Drean Commodore juntamente con una impresora y un drive en lugar de su caja registradora.

La primera ventaja que surgiría de este cambio es que no sería necesario que remarcase la mercadería una por una. Simplemente debe codificar, a su gusto, cada una de ellas.

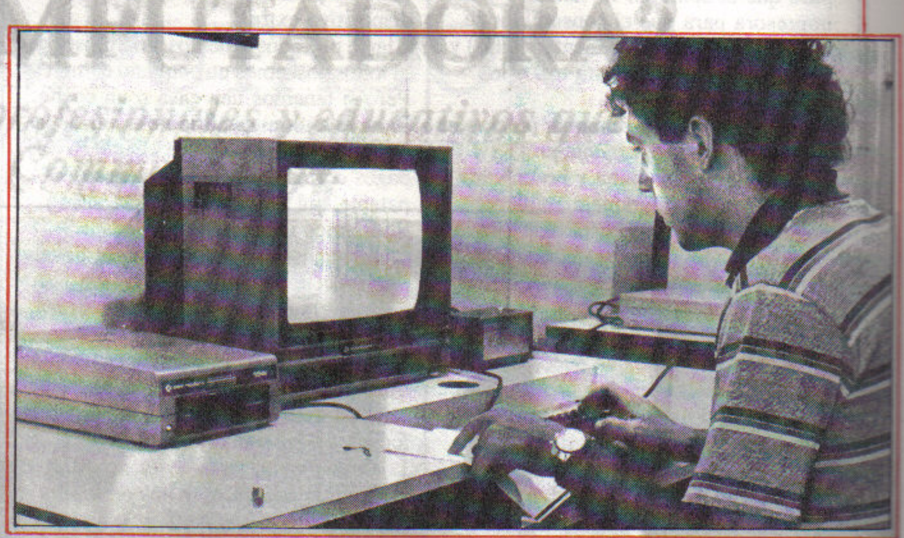
Cuando se venda un producto, el

operador debe ingresar a la máquina el código respectivo.
Esta contestará rápidamente el precio de ese producto (imprimiéndolo en pantalla para que el operador lo visualice y en impresora para que el consumidor se quede con el correspondiente recibo) ya que tendrá en la memoria todos los precios de cada código.
Por eso, a la hora de remarcar precios, se deben modificar solamente en la memoria de la máquina y no en cada producto. De esta manera se pierde menos tiempo y se ahorra más dinero.
La segunda ventaja es que, al estar codificado cada producto, se puede informar al operador cuándo una determinada mercadería alcanza un stock mínimo, para que pueda ser repuesto inmediatamente.
Además, le permite saber en cualquier instante cuál es el monto de ventas realizadas.
Al finalizar el día, el operador vuelca al disco toda la información contable y de stock de esa jornada. Este le entrega el disco al encargado, quien saca por impresora las ventas realizadas minuto a minuto, pago realizado por el consumidor, vuelto dado por el operador, total de mercadería vendida, cuántas veces se repuso el stock de un producto y, como si esto fuese poco, la cantidad de personas que entraron y salieron del local (aunque para ésto se necesitaría un dispositivo extra).
Claro, no todos tenemos un autoservice ni todos poseemos un pequeño comercio.
Pero si tenemos una casa o un coche. Para este último podríamos utilizar la Drean Commodore para visualizar los pagos correspondientes al seguro, patente, etc.
Podríamos indicarle todos los mantenimientos realizados para llevar una especie de contabilidad. Por ejemplo, la computadora nos podría pedir el kilometraje actual del coche y en base a nuestra respuesta decirnos qué mantenimiento realizar. Para nuestra casa podemos llevar la contabilidad, fechas de vencimientos, etc.
Los adolescentes y jóvenes no están al margen de las aplicaciones de la Drean Commodore. Ellos podrían utilizarla para pasar los apuntes de clase usando un editor de texto, reduciendo así el espacio que ocupa una carpeta.
Además pueden almacenar cada apunte en disco. De esta manera, si faltaron a clase, en vez de pedirle a un compañero la carpeta prestada para ponerse al día le pedirán, en cambio, el diskette prestado!
Los docentes pueden emplear la Drean Commodore para llevar las notas de cada alumno, sacar promedios, preparar los exámenes, etc.
Si tenemos una biblioteca podemos usar la Drean Commodore para que nos informe la ubicación de un determinado libro, cuántos fueron prestados, a quien, si fueron devueltos, tiempo que estuvieron ausentes, de cuántos libros de un determinado autor dispone la biblioteca, clasificarlos por temas, etc.
Si somos profesionales, por ejemplo arquitecto, utilizaremos la Drean Commodore para el cálculo de estructuras, perfiles, etc.
Puede, también, crear una determinada forma y almacenarla en el disco. Para ello se usa el lápiz óptico (un lápiz que permite dibujar sobre la pantalla del televisor).
Como vemos las aplicaciones que se le pueden dar a la Drean Commodore son muchas y variadas. Para cada profesión existe, por lo menos, una posible aplicación del equipo.



# SET DEL MICRO 6510

***En esta nota culminamos la explicación de los saltos relativos y comenzamos a describir las instrucciones del micro del Drean Commodore 64.***



La instrucción correpondiente a la dirección $4000 es BVS (Branch on Overflow Set-Saltar si hay fijación de desbordamiento) la cual se lleva a cabo si el flag V esta en "1". Nuevamente, si cuando el PC llegue a esta dirección y se verifique que este flag está seteado, el control se transferirá a:
dirección de salto = (PC) + 2 + K = $4000 + $2 + $0A = $400C
en donde se cargará el registro Y con $A1 y se suspenderá la ejecución.

| dirección | instrucción |
|---|---|
| $4000 | xx |
| $4001 | xx |
| $4002 | xx |
| $4003 | xx |
| $4004 | A2 |
| $4005 | 68 |
| $4006 | xx |
| $4007 | xx |

**SALTOS HACIA ATRAS**

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MAS SIGNIFICATIVOS | 8 - | 128 | 127 | 126 | 125 | 124 | 123 | 122 | 121 | 120 | 119 | 118 | 117 | 116 | 115 | 114 | 113 |
| | 9 - | 112 | 111 | 110 | 109 | 108 | 107 | 106 | 105 | 104 | 103 | 102 | 101 | 100 | 99 | 98 | 97 |
| | A - | 96 | 95 | 94 | 93 | 92 | 91 | 90 | 89 | 88 | 87 | 86 | 85 | 84 | 83 | 82 | 81 |
| | B - | 80 | 79 | 78 | 77 | 76 | 75 | 74 | 73 | 72 | 71 | 70 | 69 | 68 | 67 | 66 | 65 |
| | C - | 64 | 63 | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 | 49 |
| | D - | 48 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 |
| | E - | 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 |
| | F - | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| MENOS SIGNIFICATIVOS | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| MAS SIGNIFICATIVOS | 0 - | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| | 1 - | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| | 2 - | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |
| | 3 - | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 |
| | 4 - | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 |
| | 5 - | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 |
| | 6 - | 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 |
| | 7 - | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 |

**SALTOS HACIA ADELANTE**

**Tabla 2**

```
,FF00 DD 8D 07 CMP $078D,X
,FF03 DD 4C 59 CMP $594C,X
,FF06 EF       ???
,FF07 AD 95 02 LDA $0295
,FF0A 8D 06 DD STA $DD06
,FF0D AD 96 02 LDA $0296
,FF10 8D 07 DD STA $DD07
,FF13 A9 11    LDA #$11
,FF15 8D 0F DD STA $DD0F
,FF18 A9 12    LDA #$12
,FF1A 4D A1 02 EOR $02A1
,FF1D 8D A1 02 STA $02A1
,FF20 A9 FF    LDA #$FF
,FF22 8D 06 DD STA $DD06
,FF25 8D 07 DD STA $DD07
,FF28 AE 98 02 LDX $0298
```

| | |
|---|---|
| \$4008 | xx |
| | |
| \$400E | 90 ins. |
| \$400F | F4 offset |

Aquí debemos suponer que el contenido actúal del PC es \$400E.
En esta dirección se encuentra la instrucción BCC (Branch on Carry Clear-Saltar si no hay carry) la cual bifurca si el flag C esta en "0". si ello sucede, el control se transfiere a:
dirección de salto = (PC) + 2 + K = \$400E + \$2 + \$F4 = \$4004
en donde se carga el registro X con \$68. Este último ejemplo corresponde a saltos hacia atrás. Noten que el offset es un número negativo (\$F4).
Evidentemente resultaría muy tedioso efectuar el cálculo que representa la ecuación -1- cada vez que debamos calcular la nueva dirección de salto; más aún si debemos realizar un despeje de dicha ecuación ya que tendríamos que determinar el offset correcto. Para evitar esto les ofrecemos la Tabla 2 la cual representa el valor del offset acorde a la dirección que debamos saltar.

La tabla 2 representa los valores máximos para offset en saltos relativos. Verán que éstos son 127 para saltos hacia adelante y 128 para saltos hacia atrás. Además es de doble entrada ya que si damos el valor en decimal del offset, la tabla nos devuelve su correspondiente en hexadecimal y viceversa (ingreso hexa, salida decimal). En el margen izquierdo verán que hay números hexadecimales bajo el ítem "más significativos" y también bajo el ítem de "menos significativos". Por ejemplo si debemos saltar a la dirección 12 "delante" de (PC) + 2, el offset correspondiente es \$0C (primero ubicamos el 12 en la tabla de saltos hacia adelante, luego tomamos el byte más significativo -0- y luego el menos significativo -C-, formando asi el \$0C).
Si debemos saltar a la posición 75 "delante" de (PC) + 2, el offset es \$4B (nuevamente: ubicamos el 75 en la tabla de saltos hacia adelante, luego nos fijamos en el byte más significativo -4- y el menos significativo -B-, formando así el \$4B). Si, en cambio, debemos saltar a la posición 40 "detrás" de (PC) + 2, el offset es \$D8 (ubicamos el 40 en la tabla de saltos hacia atrás y nos fijamos el byte más significativo -D- y menos significativo -8-, formamos luego el \$D8). Se debe tener especial cuidado a partir de donde se empieza a contar. Siempre debe ser a partir de (PC) + 2 (recuerden que nuestra convención había determinado que (PC) corresponde a la dirección donde se encuentra el la instrucción de salto -BNE, BEQ, etc-). Es decir que debemos empezar a contar a partir de la instrucción siguiente al offset. Miren nuevamente los anteriores ejemplos para comprobar lo último dicho.

## Culminando con Saltos: la instrucción compare

Vimos que las condiciones para que los saltos se llevasen a cabo dependían del seteo o reseteo de los bit del registro de

**Tabla 1**

| CARACTER | DESCRIPCION |
|---|---|
| A | Acumulador |
| X,Y | Registros indices |
| M | Dirección de memoria |
| P | Registro de estado del micro |
| S | Registro Stack |
| V | Cambio de estado |
| / | No hay cambio de estado |
| + | Suma |
| ∧ | Función lógica AND |
| - | Resta |
| ⊻ | Función lógica OR Exclusiva |
| → | Transferencia hacia |
| ← | Transferencia hacia |
| V | Función lógica OR |
| PC | Cotador de Programa |
| PCH | Parte alta del PC |
| PCL | Parte baja del PC |
| OPER | Operando |
| # | Modo de direccionamiento inmediato. |
| ↑ | Transferencia desde el Stack |
| ↓ | Transferencia hacia el Stack |
| CO | Código de operación (en hexadecimal) |

estado del microprocesador; es decir de los distintos flags. Asimismo explicamos los saltos hacia adelante y hacia atrás suponiendo que se cumplían las condiciones respectivas (flag C=1, flag Z=0, etc.).
Las distintas instrucciones de comparación setean parte de estos flags. A través de ellas podemos determinar cuándo un número es mayor, igual o menor que otro y, dependiendo del resultado, determinar la acción a seguir. Otra de las aplicaciones que podemos darle a esta instrucción es el control de "loops" de un determinado proceso. La instrucción utiliza la unidad aritmética lógica. Básicamente la operación que realiza es (A)-(K) donde K puede ser memoria (M), registro X o Y. De acuerdo al resultado de la resta, se setearán los distintos flags.

| | N | C | Z | V |
|---|---|---|---|---|
| (A) (M) | ? | 0 | 0 | no cambia |
| (A)=(M) | 0 | 1 | 1 | no cambia |
| (A) (M) | ? | 1 | 0 | no cambia |

?= depende si el resultado es positivo o negativo
El significado de esta tabla es el siguiente:
1) Si el contenido del acumulador es menor que el contenido de una determinada posición de memoria, los flags C y Z se pondrán a cero, el flag V no cambiará y el N dependerá del resultado.
2) Si el contenido del acumulador es igual al contenido de la memoria, los flags C y Z se setearán a uno y el N a cero.
3) Si el contenido del acumulador es mayor que el contenido de la memoria, si el flag V está en "1". Nuevamente en N dependerá del resultado.
El siguiente ejemplo es un programa en código máquina que chequea si el contenido del acumulador es $62:
dddd C9 (compara ACC en forma inmediata)
dddd+1 62 (con $62)
dddd+2 F0 (BEQ-bifurca si Z=1)
dddd+3 01 (si es, vamos a la dirección dddd+5)
dddd+4 60 (finalizamos)
dddd+5 4C (saltamos a la dirección $C000)
dddd+6 00
dddd+7 C0
dddd= dirección dada.
La instrucción correspondiente a $C9 efectúa la resta que antes comentamos. En este caso se efectúa (A)−$62. Si el resultado es cero indicará que el contenido del acumulador es $62 seteándose, así, el flag Z a uno y efectuándose el correspondiente salto. En caso de que (A) sea distinto a $62 el control seguirá por la dirección dddd+4 donde finalizará.
Generalizando, toda condición que no se satisfaga producirá que el programa se siga ejecutando a partir de la dirección siguiente al offset.

Finalmente publicamos el set de instrucciones para luego explicarles cada una de ellas. Antes de comenzar, pongamonos de acuerdo con la notación a utilizar según Tabla 1.

**ADC** Suma memoria al acumulador con carry
**Operación: (A) +(M)+C →A,C**

| N | Z | C | I | D | V |
|---|---|---|---|---|---|
| ✓ | ✓ | / | / | / | ✓ |

| modo de direccionamiento | mnemotécnico | CO | Nro. de bytes | Nro. de ciclos |
|---|---|---|---|---|
| Inmediato | ADC # Oper | 69 | 2 | 2 |
| Página Zero | ADC Oper | 65 | 2 | 3 |
| Página Zero, X | ADC Oper, X | 75 | 2 | 4 |
| Absoluto | ADC Oper | 6D | 3 | 4 |
| Absoluto, X | ADC Oper, X | 7D | 3 | 4 |
| Absoluto, Y | ADC Oper, Y | 79 | 3 | 4 |
| (Indirecto, X) | ADC (Oper, X) | 61 | 2 | 6 |
| (Indirecto), Y | ADC (Oper), Y | 71 | 2 | 5 |

AND Efectua AND entre acumulador y memoria
Operación: A∧M→A

| N | Z | C | I | D | V |
|---|---|---|---|---|---|
| ✓ | ✓ | / | / | / | / |

| modo de direccionamiento | mnemotécnico | CO | Nro. de bytes | Nro. de ciclos |
|---|---|---|---|---|
| Inmediato | AND # Oper | 29 | 2 | 2 |
| Página Zero | AND Oper | 25 | 2 | 3 |
| Página Zero, X | AND Oper, X | 35 | 2 | 4 |
| Absoluta | AND Oper | 2D | 3 | 4 |
| Absoluta, X | AND Oper, X | 3D | 3 | 4 |
| (Indirecta, X) | AND (Oper, X) | 21 | 2 | 6 |
| (Indirecta), Y | AND (Oper), Y | 31 | 2 | 5 |
| Absoluta, Y | AND Oper, Y | 39 | 3 | 4 |

ASL Desplaza un bit hacia la izquierda (memoria o acumulador)
Operación: C ← 7 6 5 4 3 2 1 0 ← 0

| N | Z | C | I | D | V |
|---|---|---|---|---|---|
| ✓ | ✓ | ✓ | / | / | / |

| modo de direccionamiento | mnemotécnico | CO | Nro. de bytes | Nro. de ciclos |
|---|---|---|---|---|
| Acumulador | ASL A | 0A | 1 | 2 |
| Página Zero | ASL Oper | 06 | 2 | 5 |
| Página Zero, X | ASL Oper, X | 16 | 2 | 6 |
| Absoluta | ASL Oper | 0E | 3 | 6 |
| Absoluta, X | ASL Oper, X | 1E | 3 | 7 |

BCC Bifurca si el carry está en o
Operación: Salta si C=0

| N | Z | C | I | D | V |
|---|---|---|---|---|---|
| / | / | / | / | / | / |

| modo de direccionamiento | mnemotécnico | CO | Nro. de bytes | Nro. de ciclos |
|---|---|---|---|---|
| Relativo | BCC Oper | 90 | 2 | 2 |

BCS Bifurca si el carry esta en 1
Operación: Salta si C=1

| N | Z | C | I | D | V |
|---|---|---|---|---|---|
| / | / | / | / | / | / |

| modo de direccionamiento | mnemotécnico | CO | Nro. de bytes | Nro. de ciclos |
|---|---|---|---|---|
| Relativo | BCS Oper | BO | 2 | 2 |

# ADNUM

***A través de este entretenido juego tenemos que descubrir un número elegido por la computadora. Esta, para ayudarnos, nos indica en cada intento cuántos dígitos fueron correctos***

```
EL OBJETIVO DE ADNUM ES ADIVINAR
EL NUMERO QUE LA COMPUTADORA NOS
PREPARA, EN EL MENOR NUMERO DE
INTENTOS POSIBLES TENIENDO EN CUENTA
QUE:

1> PUEDE HABER DE 2 A 10 DIGITOS.

2> LOS DIGITOS NO SE PUEDEN REPETIR.

3> NOSOTROS ELEGIMOS CUANTOS HABRA, Y
   EN CUANTOS INTENTOS LO HAREMOS.

4> DESPUES DE CADA INTENTO TENEMOS:
   UN MENSAJE QUE INDICA EL NUMERO DE
   DE INTENTO, LOS DIGITOS A VALORAR,
   Y EL RESULTADO OBTENIDO.
   C:NUMEROS DE DIGITOS COLOCADOS
   D:DIGITOS DESCOLOCADOS PERO
     PRESENTES

 PULSA UNA TECLA PARA COMENZAR
```

El objetivo es adivinar el número seleccionado por la computadora. Nosotros podemos determinar qué cantidad de dígitos tendrá (entre 2 y 10) y en cuántos intentos, como máximo, trataremos de adivinarlo. El número que elige la computadora está formado por dígitos que no se repiten; por ello que no podemos ingresar números que tengan dos dígitos iguales. De otra manera se imprimirá un asterisco en la posición respectiva. Luego de ingresar cada intento, la computadora nos informará la cantidad de dígitos correctos (es decir aquellos que ocupan la posición correcta) y los que son correctos pero no se encuentran en la posición adecuada. Esto se representa con los caracteres C y D. Por ejemplo si la computadora seleccionó el número 710 y nosotros ingresamos 709, ocasionará C=1 (un dígito en posición correcta: 7) y D=1 (en dígito correcto pero fuera de posición: 0). El juego continúa de esa forma. Como antes mencionamos, nosotros podemos determinar el número de intentos máximos que necesitaremos para descubrir el número. Este está comprendido entre 1 y 99. En caso de llegar al número de intento sin haber descubierto el número, la computadora nos dirá el número que ella había elegido. Luego nos preguntará si deseamos seguir jugando. De todas maneras, al inicio del juego se imprimirán las instrucciones respectivas. Luego se nos preguntará por la cantidad de dígitos que tendrá el número y por el número de intentos.

```
10 REM ****************
20 REM *              *
30 REM *    ADNUM     *
40 REM *              *
50 REM ****************
52 REM INICIALIZACION
54 REM DE VARIABLES
56 REM GRAFICAS
60 POKE53281,15:POKE53280,10
62 POKE646,2
70 IC=200:FI=80:B$=CHR$(147)
72 D$=CHR$(17):R$=CHR$(29)
74 I$=CHR$(18):O$=CHR$(146)
80 NE$=CHR$(144):ED$=CHR$(187)
82 LD$=CHR$(161):RO$=CHR$(28)
84 LB$=CHR$(162)
90 EB$=CHR$(190):G$=CHR$(151)
92 IZ$=CHR$(157):AR$=CHR$(145)
94 BL$=CHR$(5)
100 HO$=CHR$(19)
110 PRINTB$D$R$R$;
```

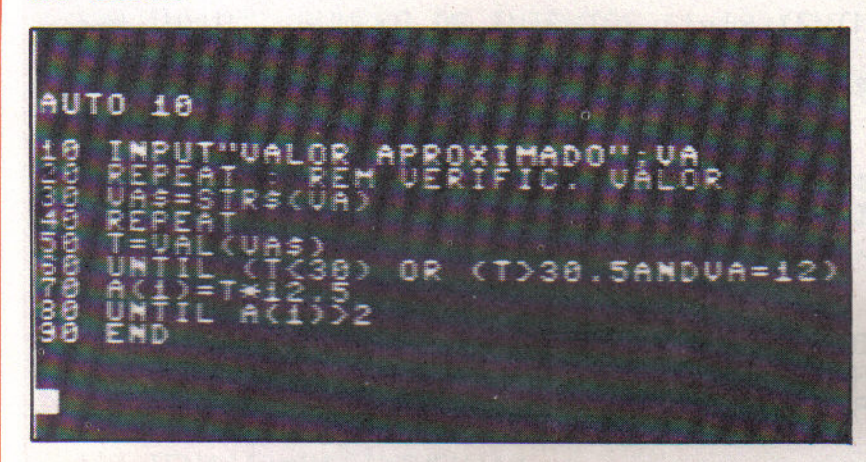
```
112 REM INSTRUCCIONES DEL JUEGO
120 PRINT"EL OBJETIVO DE "I$"ADNUM"O$" ES ADIVINAR"
130 PRINT"  EL NUMERO QUE LA COMPUTADORA NOS"
140 PRINTR$R$"PREPARA, EN EL MENOR NUMERO DE"
145 PRINTR$R$"INTENTOS POSIBLES TENIENDO EN CUENTA"
150 PRINTR$R$"QUE:"
160 PRINTD$R$R$"1> PUEDE HABER DE 2 A 10 DIGITOS."
170 PRINTD$R$R$"2> LOS DIGITOS NO SE PUEDEN REPETIR."
180 PRINTD$R$R$"3> NOSOTROS ELEGIMOS CUANTOS HABRA, Y"
```

```
190 PRINTR$R$"   EN CUANTOS INTENTOS LO HAREMOS."
200 PRINTD$R$R$"4> DESPUES DE CADA INTENTO TENEMOS:"
210 PRINTR$R$"   UN MENSAJE QUE INDICA EL NUMERO DE".
220 PRINTR$R$"   DE INTENTO, LOS DIGITOS A VALORAR,"
230 PRINTR$R$"   Y EL RESULTADO OBTENIDO."
240 PRINTR$R$"   C:NUMEROS DE DIGITOS COLOCADOS"
250 PRINTR$R$"   D:DIGITOS DESCOLOCADOS PERO"
260 PRINTR$R$"     PRESENTES"
270 PRINTD$R$R$I$"  PULSA UNA TECLA PARA COMENZAR  "
280 POKE198,0:WAIT198,1:REM GET$
290 POKE198,0:PRINTB$D$D$R$R$;
292 REM INGRESOS DE DATOS
300 INPUT"NUMERO DE DIGITOS (2-10)";N
310 IFN<2ORN>10THEN290
320 PRINTD$R$R$
330 INPUT"  NUMERO DE INTENTOS (1-99)";NT
340 IFNT<1ORNT>99THEN320
350 PRINTB$TAB(13)I$;:FORI=1TO12:PRINTCHR$(32);:NEXT:
    PRINTO$NE$ED$:POKE646,2
360 PRINT" DIGITOS:"N;TAB(13)I$"   ADNUM    "O$NE$LD$RO$;
   "INTENTOS:"NT
370 PRINTTAB(13)I$;:FORI=1TO12:PRINTCHR$(32);:NEXT:PRINTO$NE$LD$
380 PRINTTAB(14)I$;:FORI=1TO11
382 PRINTLB$;:NEXT:PRINTO$EB$G$
384 REM SELECCIONAMOS NUMERO
390 N$(1)=MID$(STR$(INT(RND(1)*10)),2)
400 FORI=2TON
410 N$(I)=MID$(STR$(INT(RND(1)*10)),2)
420 FORJ=1TOI-1
430 IFN$(I)=N$(J)THEN410
440 NEXTJ:NEXTI
450 FORT=1TONT
460 FORS=1TO37
462 PRINTCHR$(32);:NEXT
464 FORS=1TO20:PRINTIZ$;:NEXT
466 PRINTCHR$(113);
470 POKE198,0
480 FORK=1TON
490 WAIT198,1
500 GETP$(K)
```

```
510 FORJ=K-1TO1STEP-1
520 IFP$(K)=P$(J)THENPRINT"*"IZ$;:GOTO490
530 NEXTJ
540 IFP$(K)=CHR$(20)THENIFK>1THENPRINTIZ$CHR$(32)IZ$;:K=K-1
    :GOTO490
550 IFP$(K)<"0"ORP$(K)>"9"THEN490
560 TP=50:GOSUB800:PRINTP$(K);
570 NEXT
580 FORI=1TON
590 IFN$(I)=P$(I)THENM=M+1
600 NEXT
610 FORI=1TON
620 FORJ=1TON
630 IFN$(I)=P$(J)THENH=H+1
640 NEXTJ:NEXTI
642 PRINT:M$=MID$(STR$(M),2)
644 H$=MID$(STR$(H),2)
650 T$=MID$(STR$(T),2)
655 IFT<10THENT$="0"+T$
660 T$=T$+CHR$(32)
670 PRINTAR$"INTENTO:"T$;"NUMERO";
672 FORI=1TON:PRINTTP$(I);:NEXT
680 PRINT" C:";M$;" D:";H$:IFN=MTHEN730
690 M=0:H=0
695 M$="":H$=""
698 REM SI ES MULTIPLO DE 10 SUBIMOS
700 IFT/10=INT(T/10)THENPRINTHO$D$D$D$D$;
710 NEXTT:PRINTNE$"SE PASO !! ERA EL:";:FORI=1TON:PRINTN$(I);
    :NEXT:TP=5
720 PRINT" LO SIENTO":GOSUB800:GOTO750
730 TP=-5:IC=50:FI=250:GOSUB800
740 PRINTBL$"!PERFECTO! ERA EL: ";:FORI=1TON:PRINTN$(I);:NEXT
    :PRINT
750 PRINTTAB(12);"¿DESEA JUGAR"
755 PRINTTAB(12);"OTRO PARTIDO (S/N)
760 POKE198,0:WAIT198,1:GETA$
770 IFA$="S"THENRUN
780 IFA$="N"THENEND
790 GOTO760
799 REM SUBRUTINA SO.GRA
800 S=54272:VM=S+24
805 FORJ=STOVM:POKEJ,0:NEXT
810 POKE54290,16:POKES,5
812 POKES+2,128:POKES+3,7
814 POKES+5,15:POKES+6,240
820 POKES+14,5:POKES+16,128
822 POKES+17,7:POKES+19,15
824 POKES+20,240:POKES+24,15
830 FORI=ICTOFISTEP-TP
840 POKE54276,21
850 POKE54273,I:POKE54287,96
852 NEXT
860 POKE54276,16
862 FORJ=0TO50:NEXT
864 RETURN
```

# MANEJO DE LAS VARIABLES

***Siguiendo con el comentario de este lenguaje, verán que una misma variable puede ser utilizada en distintos sectores sin que se pierdan los valores iniciales de ésta.***




Comenzaremos explicando las facilidades que tiene el Simons' para el manejo de variables. Nos permite establecer valores diferentes para una misma variable. Esto es muy útil para usar la misma variable, con distintos valores, en secciones distintas de nuestro programa. De esta manera ahorramos variables y, por lo tanto, memoria. Las sentencias LOCAL y GLOBAL nos permiten realizar lo último . Con la primera de ellas (LOCAL) podemos definir nuevos valores de una o varias variables que se utilizarán localmente en una rutina.

Sentencia: LOCAL

Formato: LOCAL variable1, variable2,... donde variable1, variable2,..., son las variables locales.

Un ejemplo de esta sentencia es el siguiente:

```
10 REM EJEMPLO SENTENCIA
   LOCAL
20 A$="VALOR INICIAL":A%=
   1:A=2.5
30 LOCAL A$,A%,A
40 A$="NUEVO VALOR":A%=3:
   A=4.5
50 PRINT A$,A%,A
60 STOP
```

Cuando ejecutemos este programa en la línea 50 se imprimirán los siguientes valores:

NUEVO VALOR 3 4.5

Los valores anteriores de A$,A% y de A fueron temporalmente guardados por el Simons' en un área de memoria (similar al stack) hasta que una sentencia GLOBAL los restablezca.

Sentencia: GLOBAL

Formato: GLOBAL

Función: Restablece los valores de las variables locales.

Esta sentencia origina que todas las variables que fueron definidas con otros valores a través de la sentencia LOCAL tengan sus valores originales. Volviendo al ejemplo anterior, el uso de GLOBAL sería:

```
10 REM EJEMPLO SENTENCIAL
   LOCAL
20 A$="VALOR INICIAL":A%=1:
   A=2.5
30 LOCAL A$,A%,A
40 A$="NUEVO VALOR":A%=3:
   A=4.5
50 PRINT A$,A%,A
60 GLOBAL
70 PRINT A$,A%,A
80 STOP
```

En la línea 50 se imprimen los mismos valores que en el ejemplo anterior, es decir NUEVO VALOR, 3 y 4.5. En la línea 60 regresamos a las variables involucradas sus valores originales antes de producirse una sentencia LOCAL. Esto lo efectuamos a través de la sentencia GLOBAL. Finalmente en la línea 70 imprimimos los valores iniciales, es decir:

VALOR INICIAL 1 2.5

## Manejo de errores en el Simons'

Otra de las grandes características que tiene el Simons' es la posibilidad de transferir el control del programa en caso de ocurrir algún tipo de error, ya sea de sintaxis o de algún ingreso externo como ser las asignaciones de string a una variable de punto flotante. Pasemos pues a comentar la:

Sentencia: ON ERROR

Formato: ON ERROR:GOTO número de línea.

Función: Esta sentencia transfiere el control del programa al número de línea especificado en él cuando se produce un error en la ejecución del programa. Además, setea en las variables ERRLN y ERRN en qué número de línea ocurrió y el código del error. Este puede ser:

| CODIGO DE ERROR | DESCRIPCION |
|---|---|
| 1 | TOO MANY FILES |
| 2 | FILE OPEN |
| 3 | FILE NOT OPEN |
| 4 | FILE NOT FOUND |
| 5 | DEVICE NOT PRESENT |
| 10 | NEXT WITHOUT FOR |
| 11 | SINTAX |
| 12 | RETURN WITHOUT GOSUB |
| 13 | OUT OF DATA |
| 14 | ILLEGAL QUANTITY |
| 15 | OVERFLOW |
| 16 | OUT OF MEMORY |
| 17 | UNDEFINED STATEMENT |
| 18 | BAD SUBSCRIPT |
| 19 | RE-DIMENSIONED ARRAY |
| 20 | DIVISION BY ZERO |
| 21 | ILLEGAL DIRECT |
| 22 | TYPE MISMATCH |
| 23 | STRING TOO LONG |

Un ejemplo de la utilización de esta sentencia es el siguiente programa:

```
10 REM EJEMPLO DE ON
   ERROR
20 ON ERROR:GOTO 500
30 OPEN2,8,2,"MAESTRO,S,W"
40 PRINT#2,"ARCHIVO
   MAESTRO"
50 OPEN3,8,3,"MAESTRO,S,W"
60 PRINT#3,"EOF"
70 CLOSE2
```

```
80 CLOSE3
90 STOP
500 IF ERRN=2THENPRINT "EL
    ARCHIVO YA SE
    ENCUENTRA ABIERTO.
    ERROR EN LA
    LINEA";ERRLN
510 STOP
```

Como observarán abrimos el mismo archivo dos veces (primero con el número 2 y luego con el número 3-líneas 30 y 50). El Simons' detectará el error y saltará a la línea especificada en el ON ERROR, es decir a la línea 500. Aquí se chequea por el código de error número 2 (File Open). En caso que éste ocurra, se imprimirá el mensaje correspondiente informando el error juntamente con el número de línea donde ocurrió.
Juntamente con esta sentencia se encuentra el comando OUT y el NO ERROR los cuales desactivan la más reciente sentencia ON ERROR. De esta manera la C-64 retorna a su manejo normal de errores.

## Funciones numéricas

El Simons' dispone también de nuevas funciones orientadas a operaciones aritméticas tales como división entera, conversión de números, etc.
Comando: MOD
Formato: MOD(X,Y)
Función: Este comando retorna el resto de la división de X por Y (X e Y deben ser números enteros). El comando puede ser utilizado en modo directo o dentro de un programa. Por ejemplo:
MOD(10,3)
retornará el valor 1
Comando: DIV
Formato: DIV(X,Y)
Función: Retorna el cociente entero entre dos números de punto flotante o reales. Por ejemplo:
DIV(10,3)
retornará el valor de 3. Este comando, al igual que MOD, puede utilizarse en modo directo o dentro de un programa.
Comando: FRAC
Formato: FRAC(X)
Función: Retorna la parte fraccionaria del número representado por X. Por ejemplo:
FRAC(12.9921)
retornará la fracción .9921. También se puede utilizar esta función en modo directo o en un programa.
Para la conversión de números, el Simons' dispone del operador "%" que, convierte números binarios en decimales, y el operador "$" que transforma números hexadecimales en decimales. Ejemplo de ello sería:
PRINT %1111
el cual imprime el número decimal 15.
Para el caso de conversión hexadecimal:
PRINT $C000
el Simons' imprimirá el número decimal 49152. Se permite realizar combinaciones de estas funciones como por ejemplo:
PRINT $C000+%1111
Aquí retornará el número 49152+15=49167
Otras de las funciones que dispone el Simons' es la OR exclusiva:
Comando: EXOR
Formato: EXOR(X,Y)
Función: Efectúa la OR exclusiva entre los números representados por X e Y.
Esta función comúnmente se representa como X⊕Y. La tabla de verdad correspondiente a ella es:

| X | Y | X⊕Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Por ejemplo EXOR(10,7) se efectúa como:
10=1010
7=0111
el resultado es: 13=1101

# TRUCOS

## *EL BUFFER DEL TECLADO*

El CHR$ (131) es equivalente a oprimir la teclas SHIFT/STOP, que efectúa la carga y ejecución de un programa almacenado en cassette. Si ustedes generan este código verán que no sucede lo descripto. La única manera de simular ésto es tipeando: 10 POKE 631, 131: POKE 198,1: END
Ejecutando esta línea se imprimirá el mensaje de LOAD y se efectuará la carga correspondiente.

## REDIMENSIONAMIENTO DE VECTORES

Es sabido que cuando redimensionamos vectores (más de una sentencia DIM en nuestro programa) el intérprete Basic detiene la ejecución imprimiéndonos REDIM'D ARRAY
Con la sentencia que aquí les ofrecemos podrán redimensionar conjuntos sin interrumpir el normal funcionamiento del programa

```
POKE49, PEEK(47)
POKE50,PEEK (48)
```

Estas líneas deben ponerse antes de la próxima sentencia DIM

## *MEMORIA EN PANTALLA*

En ocasiones necesitamos determinar cuales áreas de memoria tienen una determinada serie de datos. A través de este utilitario podremos imprimir en la parte superior de la pantalla un área de memoria. Las que pueden imprimirse van de la 0 hasta la 255. El programa imprime todas las instrucciones para utilizarlo correctamente. Una vez activado solo debemos tipear POKE923, N, (donde N es el área a imprimir), para visualizar ese "segmento" de memoria. El programa dispone de un check sum para comprobar que los valores de los DATA fueron tipeados correctamente. Para desactivarlo debemos oprimir la tecla de RESTORE y STOP simultáneamente. Para activarlo solo basta con SYS900 (el programa reside el buffer del cassette).

```
1  REM MEMORIA EN
   PANTALLA
3  DATA 120, 169, 145, 141, 20, 3,
   169, 3
4  DATA 141, 21, 32, 88, 96, 173,
   136, 2
5  DATA 141, 158, 3, 162, 0, 189,
   0,0
6  DATA 157, 0, 4, 202, 208, 247,
   162, 0
7  DATA 173, 134, 2, 157, 0, 216,
   202, 208
8  DATA 250, 76, 49, 234, 4764
10 FORJ=1 TO44: READK:
   CS=CS+K:NEXT: READCK
20 RESTORE: IFCS<>
   CKTHENPRINT "ERROR EN
   DATAS": STOP
30 FORJ=900TO943: READK:
   POKEJ,K: NEXT
40 X$="RESTORE"
50 PRINT TAB (18) "POKE923,N--
   SYS900"
60 PRINT "POKE923,N PARA
   OBSERVAR LA PAGINA DE
   MEMORIA N"
70 PRINT"OPRIMA "X$" Y STOP
   PARA DESACTIVAR EL
   UTILITARIO"
80 PRINT "PARA ACTIVARLO
   TIPEE SYS900"
90 SYS900: PRINT "UTILITARIO
   ACTIVADO"
95 NEW
```

# HESMON-MONITOR

***Rating Total: A***
***Creatividad: A***
***Documentación: B+***
***Valor en relación al costo: A−***
***Computadora: Drean Commodore 64***
***Editor: Peek***

```
AC000  A9 FF       LDA #$FF
AC002  A2 FF       LDX #$FF
AC004  20 D2 FF    JSR $FFD2
AC007  A9 23       LDA #$23
AC009  20 D2 FF    JSR $FFD2
AC00C  24 23       BIT $23
AC00E  D0 F0       BNE $C000
AC010  90 1C       BCC $C02E
AC012  18          CLC
AC013  20 AA EF    JSR $EFAA
AC016  00          BRK
AC017
```

**HESMON es un potente monitor diseñado para la C-64. Este utilitario está escrito íntegramente en lenguaje máquina y ocupa muy poca memoria (aproximadamente 4 Kb). Nos permite trabajar en código de máquina o en mnemotécnicos. La única restricción que dispone el HESMON para trabajar en este modo es que no se permiten saltos hacia labels o etiquetas. En realidad todos los monitores, incluyendo el que acompaña al Macro Assembler, no soportan estos tipos de saltos. Aquí, al igual que en el Macro, se debe ingresar la dirección de salto explícitamente. La ventaja de trabajar con saltos implícitos, implica trabajar con direcciones simbólicas, desentendiéndose, así, del cálculo del OFFSET tan necesario cuando se programa en lenguaje máquina. De todas maneras el HESMON amortigua un poco este procedimiento ya que suministra comandos para poder realizar sumas y restas en hexadecimal, conversión decimal-hexa y conversión hexa-decimal. Otra de las desventajas que tiene este utilitario es que no permite constantes en el sistema de numeración binario u octal; sólo permite constantes hexadecimales. De esta manera si ésta es ingresada anteponiendo el signo '$' se entenderá que se trata de una constante hexadecimal. Debido a que este es el único tipo permitido, el signo '$' es opcional.**

**Suministra al programador un total de 29 comandos. Estos son según el cuadro 1: Al comenzar, el monitor nos imprime los contenidos de los registros de trabajo. Ellos son: PC (contador de programa), IRQ (interrup request vector), SR (status register), AC (acumulator), XR (X register), YR (Y register), SP (stack pointer). Los contenidos iniciales de éstos son:**

| PC | IRQ | SR | AC | XR | YR | SP |
|---|---|---|---|---|---|---|
| ;0000 | EA31 | 27 | 00 | 00 | 00 | FA |

**Nuestros programas se pueden escribir a partir de la dirección $0801 hasta $9FFF y a partir de $C000 hasta la dirección $CFFF.**

| Comando | Descripción |
|---|---|
| A | Ingresa una línea en assembler |
| B | Efectúa un Breakpoint |
| C | Compara bloques de memoria |
| D | De-ensambla |
| E | Compatibiliza el programa actual para que pueda ser ejecutado por otra computadora |
| F | Llena el área de memoria especificado con el byte seleccionado |
| G | Ejecuta el programa partir de la dirección especificada |
| H | Localiza un o una serie de bytes dentro del rango de memoria seleccionado |
| I | Interpreta la memoria |
| L | Carga un programa desde el periférico seleccionado |
| M | Imprime los contenidos de las direcciones deseadas |
| N | Mueve programas cambiando sus posiciones relativas de salto |
| O | Copia o almacena en impresora o en disk la pantalla actual |
| P | Copia en impresora la pantalla actual |
| Q | Realiza un rápido TRACE |
| R | Imprime los contenidos de los registros del 6510 |
| S | Graba en el periférico seleccionado el programa ubicado en las direcciones ingresadas |
| T | Transfiere bloques de memoria |
| U | Chequea la RAM color |
| V | Verifica RAM |
| W | Realiza una ejecución paso a paso del programa |
| X | Retorna al Basic estandar |
| # | Convierte decimal a hexa |
| $ | Convierte hexa a decimal |
| + | Suma hexadecimal |
| − | Resta hexadecimal |
| : | Modifica contenidos de direcciones |
| ; | Modifica los registros de trabajo |
| , | Modifica el de-ensamblador |

cuadro 1

El HESMON es muy potente en búsqueda de contenidos dentro de una zona de memoria. No sólo puede determinar la dirección donde se encuentra el byte buscado sino que, además, puede localizar una secuencia de caracteres.

Cuando se trabaja con el comando 'A', el HESMON opera como un intérprete ya que luego de ingresar nuestra instrucción en mnemotécnico, éste la traduce, e imprime el código de máquina con nuestra instrucción.

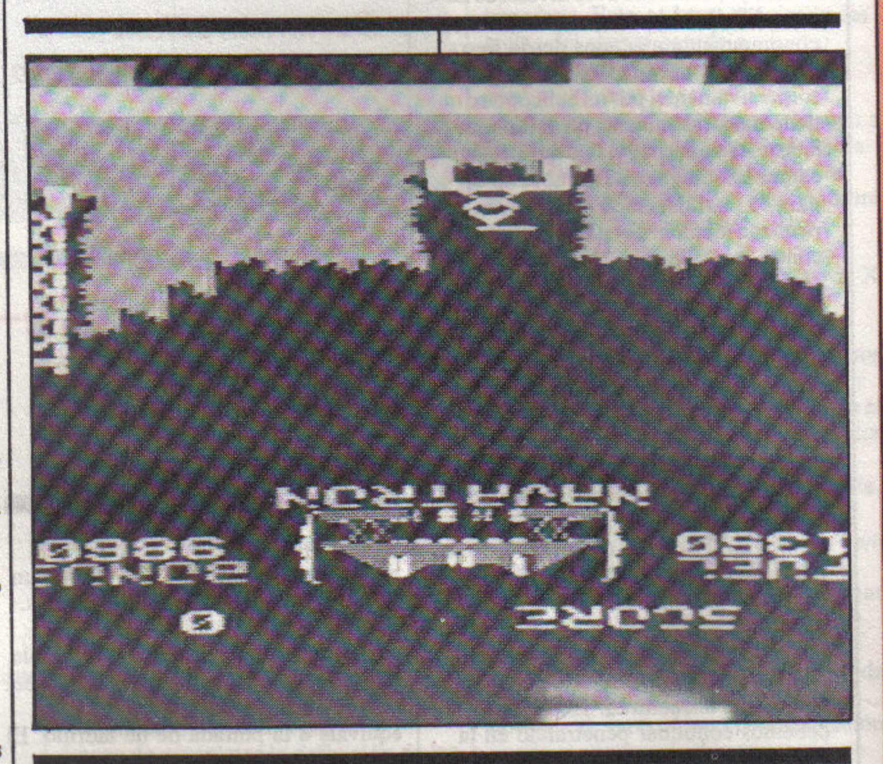
Otro comando potente es el save (S) quien nos permite grabar secciones de memoria en el periférico seleccionado. El comando M lista los contenidos de las direcciones seleccionadas en formato de líneas de ocho bytes, mientras que el comando I lista los códigos ASCII que ellas representan. A través de éste podemos localizar los mensajes de error que utiliza el sistema operativo de la C-64.

Algo importante de resaltar es su potente editor de pantalla. Por ejemplo, si oprimimos la tecla que mueve el cursor hacia abajo luego de solicitar un comando M, se nos listará los contenidos que corresponden a las direcciones siguientes en donde finalizó el comando (hacia direcciones superiores). Si oprimimos la tecla que mueve el cursor hacia arriba, nos moveremos entre las direcciones inferiores a la última ejecutada por el comando M. Es decir que con esas dos teclas podemos desplazarnos por toda la memoria del C-64.

Hay detalles que en este utilitario sí están contemplados como ser el pasaje de la dirección $FFFF a la $0000, cuando se supera la primera por algún motivo.

Los programadores de lenguaje máquina podrán encontrar en el HESMON una gran ayuda para desarrollar sus programas.

# FORT APOCALYPSE



*Rating Total: C*
*Creatividad: B*
*Documentación: C*
*Profundidad del juego: A*
*Desafío: Difícil*
*Gráficos: B*
*Valor en relación al costo: B*
*Mantiene el interés: A*
*Computadora: Drean Commodore 64*
*Editor: Peek*

# MINER

***Rating Total: C***
***Creatividad: C***
***Documentación: C***
***Profundidad del juego: C***
***Desafío: Difícil***
***Gráficos: C***
***Valor en relación al costo: B***
***Mantiene el interés: B+***
***Computadora: Drean Commodore 64***
***Editor: Peek***

Este simpático minero debe ir recorriendo los distintos niveles de un túnel esquivando difíciles obstáculos. Su objetivo es "pintar" toda la excavación en el menor tiempo posible. Esto se logra a través de la caminata del minero en la mina; un paso de él equivale a la pintada de un ladrillo. El túnel está formado por pendientes, escaleras y peligrosas cornisas que nuestro amigo debe hábilmente saltar. El recorrido por el túnel termina cuando él pinte todos los ladrillos de éste. Además debe esquivar unos raros animales subterráneos (conocidos como Rarex), los cuales pueden destruír al minero. A ellos también se los puede destruír. Para ello se debe esperar a que su forma pase de "partida" a "entera". Aquí simplemente basta con chocarlos para que dejen de existir.

El primer túnel está formado por rampas y escaleras con cuatro Rarex ubicados en el anteúltimo y primer nivel. El minero comienza su recorrido en el último nivel (el más bajo). Debe acceder a la primera rampa a través de un salto, el cual se logra oprimiendo el botón de disparo del joystick. Debe seguir subiendo y pintando todos los ladrillos sin que ningún Rarex lo mate.

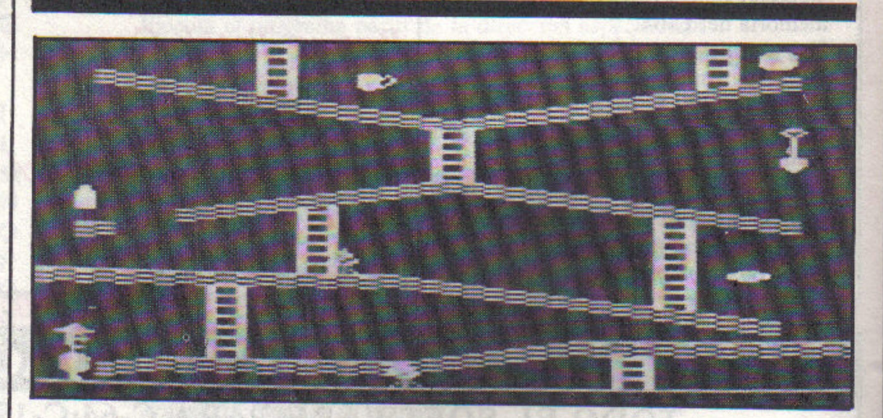
Además dispone de un cierto tiempo para completar la tarea. En caso de que el contador llegue a cero antes de culminar con ella, el minero sufrirá el mismo efecto que si hubiese chocado con un Rarex partido.

El segundo túnel está formado por escaleras y toboganes y, por supuesto, los inseparables Rarex. Nuevamente debemos pintar todos los ladrillos antes de que se nos acabe el tiempo y de no caer en un tobogán ya que éste nos llevaría al comienzo del recorrido.

El tercer túnel está constituído por escaleras, un ascensor y Rarexs. La única diferencia que hay aquí es que los ascensores son en realidad teletransportadores (similares a los de la nave ENTERPRISE). Gracias a él accedemos a los distintos niveles sin correr riesgo de choque con los Rarexs. A partir del cuarto túnel se complica el juego. Aquí prácticamente no hay más escaleras. El minero debe ir saltando a través de pequeñas bases (las cuales también se pintan) para cumplir con la tarea y poder así pasar al túnel siguiente.

Este juego es realmente entretenido e interesante. Los movimientos del minero son comandados simplemente por el manejo adecuado del joystick.

Podrán encontrar en él la emoción e intriga que en otros juegos no hallaron.



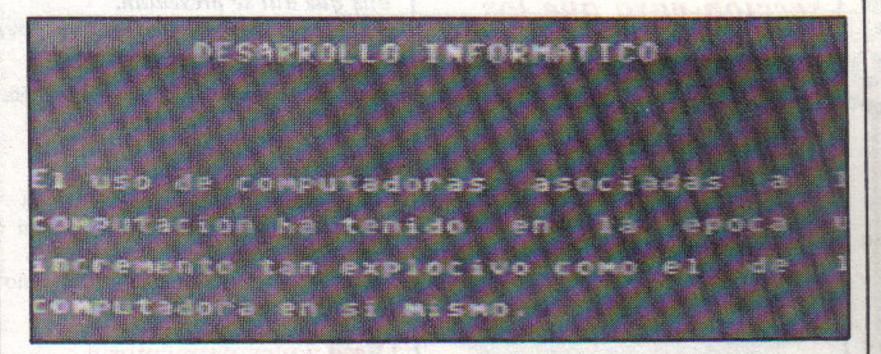
El objetivo de este interesante juego es rescatar 16 personas las cuales se encuentran dispersas en los distintos niveles de una fortaleza subterránea.

Para ello, nosotros comandamos un helicóptero, a través de joystick, el cual es capaz de disparar misiles y bombas.

Desde ya, el acceso a los niveles se hace muy difícil debido a que somos continuamente atacados desde tierra por tanques enemigos y, sorpresivamente, por un helicóptero que tiene nuestras mismas posibilidades de fuego. Al comienzo del juego debemos cargar los tanques de combustibles, para lo cual debemos realizar una pequeña maniobra evitando estrellarnos contra la plataforma. No debemos olvidarnos de la gravedad existente (elegida por el jugador), la cual provoca una atracción de nuestra nave hacia la tierra. Luego del abastecimiento de combustible estamos listos para emprender la misión. Primero hay que ingresar al nivel I. Para acceder a éste tenemos que destruir una barricada que obstruye la entrada y, al mismo tiempo, evitar que los tanques enemigos nos destruyan. Para ello disponemos de un mapa que nos indica nuestra posición dentro de NAVATRON y las de los tanques enemigos en la fortaleza. Una vez adentro, tenemos que rescatar un número no determinado de personas que allí se encuentran. Hecho esto, debemos continuar penetrando en la fortaleza. El acceso al nivel II se realiza esquivando rayos láser enviados por el enemigo. En este nivel podemos reabastecernos de combustible, tarea nada fácil debido a que se nos ataca para que no podamos llegar al tanque de combustible. Por suerte disponemos del mapa de acción para guiarnos dentro de la fortaleza subterránea. Esta tiene una forma circular, ya que a medida que avanzamos volvemos a pasar por lugares donde estuvimos. Otro de los rayos que tiene el fuerte son los rayos transportadores, los cuales nos trasladan a distintos sectores de la fortaleza, inclusive a otro nivel.

El acceso al nivel III se realiza a través de un elevador, el cual está muy protegido. Si logramos llegar hasta él, veremos cómo comenzamos a descender al próximo nivel aumentando así, la dificultad del juego. En cierta forma ésta se ve en ocasiones disminuída ya que los tanques destruyen todo lo que esté volando, incluyendo su propio helicóptero. Esto lo debemos aprovechar para continuar el avance y poder destruír al temible Agtrox.

# EASY SCRIPT




***Rating Total: A***
***Creatividad: A***
***Documentación: B+***
***Valor en relación al costo: A—***
***Computadora: Drean Commodore 64***
***Editor: Peek***

Este utilitario es un potente procesador de texto diseñado para la C-64 juntamente con una impresora compatible a él. Al comienzo del programa se nos interroga sobre la pantalla en dónde vamos a trabajar, es decir el número de columnas máximo, el cual está comprendido entre 40 y 240 columnas. Si oprimimos la tecla RETURN setearemos este valor, por default, a 40. Si en cambio trabajamos en, por ejemplo, 80 columnas, a medida que vamos editando el texto, y cuando éste supere la columna 40, se realiza automáticamente un desplazamiento (scroling) horizontal hasta llegar a la columna 80.

Luego se nos pide que ingresemos el dispositivo externo de almacenamiento. Este puede ser cinta (Tape) o diskette (Disk) en donde almacenaremos el texto editado. Por default se asume diskette.

Finalmente el programa requiere el tipo de impresora que utilizaremos. Estos están codificados de 0 a 4 siendo sus equivalencias:

0=CBM
1=MX80
2=SPINWRITER
3=QUME/DIABLO/8300
4=OTRAS

La primera corresponde a impresoras Commodore, la segunda a impresoras tipo Epson, la segunda y tercera a las impresoras que allí se describen. Por último la quinta representa a otras impresoras no descriptas aquí. En el caso de optar por ella, el editor preguntará las características de nuestra impresora, es decir si es RS232, serie o centronics. En caso de ser la primera, el utilitario nos pedirá definir el registro de control y el registro de comando. Aquí debemos ingresar un número entre 0 y 255. El manual de Easy Script nos explica detalladamente cómo se establecen dichos valores. Ellos están en función de la velocidad de transmisión, paridad, etc.

Luego de completar la definición de los dispositivos que utilizaremos, estamos en condiciones de comenzar a editar nuestro texto. En la pantalla aparecerá la línea de estado quien lleva toda la información necesaria sobre nuestra edición y, también, el número de línea y de columna donde se encuentra actualmente el cursor. Esta línea nos indica, por ejemplo, si estamos en mayúsculas a través de una C (del inglés capitals), si estamos en modo decimal, búsqueda y sustitución, modo disk, tabuladores, transferencia de bloques de texto, etc. Con respecto a la edición, el EASY SCRIPT permite definir margen izquierdo y derecho, justificación, centrado, encabezamientos y pies de página, espacios entre las líneas impresas, tabuladores horizontales y verticales, longitud de página y longitud de texto. La tecla de función 3(F3) se utiliza para ingresar los diversos comandos de impresión (los últimos descriptos). Cuando ésta se oprime aparece en la pantalla un asterisco en campo inverso el cual representa la aceptación de los comandos antes citados. Otras de las funciones que dispone este procesador es la de transferencia o copia de bloques de texto. Por ejemplo si al terminar de escribir un determinado texto nos dimos cuenta que la posición que éste ocupa no es la mejor, es decir debemos cambiarlo de lugar, simplemente oprimimos F1 y luego la tecla R. En la línea de estado aparecerá el mensaje Set Range. De ésta forma el editor nos pide que definamos el subtexto a copiar o transferir. Esto se logra moviendo las teclas del cursor acorde a nuestro requerimiento. A medida que ésto ocurre el subtexto es resaltado en campo inverso. Finalizamos el rango a copiar/transferir presionando la tecla RETURN. Seguidamente debemos posicionar el cursor en la zona donde irá el subtexto. Una vez hecho esto simplemente oprimimos la tecla F1 seguida de A (en caso de copia) o de X (en caso de transferencia).

El EASY SCRIPT también nos permite efectuar búsqueda de subtextos y sustituirlos por el seleccionado por nosotros o simplemente localizarlo en el texto. Otra de las ventajas de este editor es que podemos visualizar el texto confeccionado antes de sacarlo por la impresora contando con todas las funciones para que dicha visualización sea lo más cómoda. Por ejemplo si seteamos los tabuladores para trabajar en ochenta columnas, la pantalla no llega a representar todo el ancho del texto. Utilizando las diversas teclas de función podemos realizar scroling horizontal, vertical, etc. Esta impresión puede realizarse de dos formas: continua o no continua. La primera muestra todo el texto mientras que la segunda realiza lo mismo con la salvedad de que cuando termina cada página espera que nosotros oprimamos la tecla C para indicarle que continúe con la impresión. Desde ya EASY SCRIPT permite borrar fácilmente el texto editado. Este se puede realizar borrando todo el texto, hasta el próximo punto de donde se encuentra el cursor o seleccionando el texto a borrar a través del cursor. Al moverlo, el texto se pondrá en campo inverso mostrando así lo que se borrará. Esto se llevará a cabo cuando se oprima la tecla RETURN.

Otra de las características importantes es que nos permite definir nuestros propios caracteres como ser, por ejemplo, la ñ la cual no está definida en el set de caracteres de la lengua inglesa.

Las características definidas aquí son simplemente algunas de las tantas que dispone el EASY SCRIPT, uno de los más potentes editores de texto para la C-64 que actualmente se encuentra en nuestro mercado. Viene acompañado de su correspondiente manual

# CORREO - CONSULTAS

## Drean Commodore

***Tengo una Drean Commodore 64 con disketera 1541 y una impresora MPS803. Quisiera saber si es posible imprimir en la impresora lo que se encuentra en pantalla.***
***Les rogaría que me detallen paso a paso el procedimiento para efectuar ésto.***
***Muchas gracias.***
***Alejandro Río, Capital***

Es posible sacar por impresora lo que se encuentra actualmente en pantalla. Para ello debemos recorrer las direcciones de memoria correspondientes a la zona de pantalla e imprimir los contenidos que allí se encuentran. Cada caracter que visualizamos está codificado en una norma llamada ASCII (American Standar Code for Information Interchange-Código americano normalizado para el intercambio de información). Por ejemplo la letra "A" se codifica como 65, la letra "B" 66, la "C" 67, etc. Los contenidos que antes mencionamos se referían a estos valores. Por ejemplo si en la línea superior de la pantalla tenemos los caracteres ABC, los contenidos de las primeras tres direcciones de memoria (siempre correspondiente a la pantalla) serán 65,66, y 67. Esta zona está comprendida desde la dirección 1024 a la 2023 (ambas decimales). La primera línea de pantalla está formada por las direcciones 1024,1025,1026,...,1063. La segunda por las direcciones 1064,1065,1066,...,1103. La tercera por 1104,1105,1106,...,1143. Cada nueva línea se forma sumando 40 (total de columnas) al comienzo de la anterior (el inicio de la segunda línea de pantalla es 1024+40=1064, la tercer línea como 1064+40=1104, etc.). De todas maneras mira el manual que acompaña a tu computadora para completar el concepto.
Para finalizar debemos ir, entonces, recorriendo las direcciones desde la 1024 hasta la 2023.
El siguiente programa saca por impresora lo que actualmente se encuentra en pantalla. Puedes utilizarlo como una subrutina. Es decir que cuando quieras usarlo, debes hacer GOSUB60000:
Nota: Este programa no puede imprimir caracteres en video inverso.

***Continuamos con esta sección para que los lectores planteen sus consultas y sugerencias. Para eso deberán escribir a nuestra redacción: Cerrito 1320, (1010), Buenos Aires***

## Mapa de memoria

***Ante todo quiero felicitarlos por la revista que ustedes editan. Es la única en su tipo que sale a cubrir los requerimientos de los usuarios del Drean Commodore.***
***Les pido una aclaración respecto a la nota de MAPA DE MEMORIA del número 3. No comprendo con qué está relacionado el título CONSTITUCION DEL DATA DIRECTION REGISTER juntamente con la descripción de los bits que allí se presentan.***
***Carlos Matti - Buenos Aires***

**Drean Commodore**
La descripción de cada uno de los bits que se encuentran bajo el título "CONSTITUCION DEL DATA DIRECTION REGISTER" corresponde a la dirección $0000. Si relees la nota verás que esa dirección representa a este registro, por lo que creímos conveniente explicar cada uno de los bits que lo constituyen.

## Direcciones de memoria

***Les escribo con el fin de que me expliquen qué son y para qué sirven las direcciones de memoria del Drean Commodore.***
***Mabel Cunas - Capital***

**Drean Commodore**
Las direcciones de memoria, no sólo del Drean Commodore, sino de cualquier computadora, son como "estantes" donde se guardan los datos que permiten el buen funcionamiento de la computadora. Cuando escribimos un programa, éste se va almacenando en esos "estantes". Cuando lo ejecutamos la computadora necesita de los datos que hay en otros "estantes" para poder realizar esa tarea.

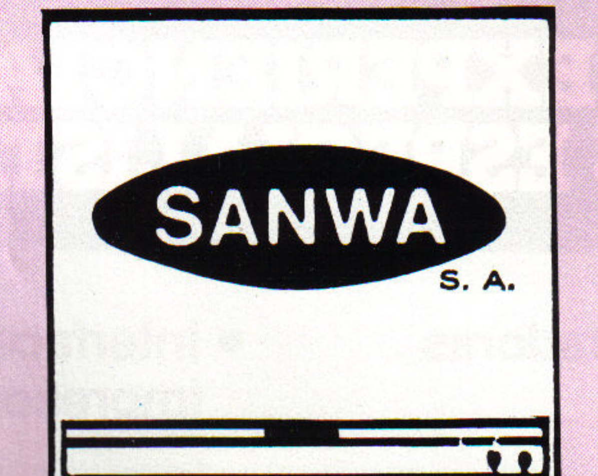
## Aclaración:

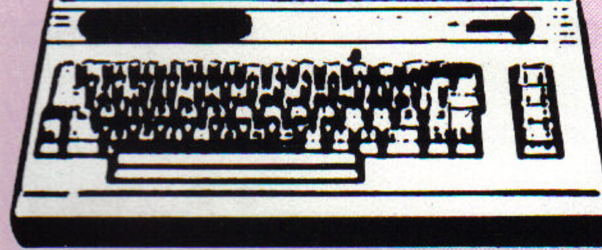
En la nota de ASSEMBLER 3ª parte (número 3, pág. 23) hay un subtexto que no tiene título y que se confunde con la nota en sí. Este comienza con "Coloque con la dirección $5FA1 el valor de $FF. Luego escriban...". Nuestro objetivo era que ésto fuese un ejercicio para que ustedes lo realicen. Ahora ya lo pueden hacer.

```
60000 REM RUTINA DE IMPRESION
60010 PO$=CHR$(16):QT$=CHR$(34)
60020 RV$=CHR$(18):RO$=CHR$(146)
60030 VR=PEEK(648)*256
60040 OPEN4,4
60050 FOR CL=0 TO 24:QF=0:AS$=""
60060 FOR RO=0 TO 39
60070 SC=PEEK(VR+40*CL+RO)
60080 IFSC=34THENQF=1-QF
60090 IFSC<>162THEN60120
60100 QF=1-QF:IFQF=1THENAS$=AS$+RV$+QT$:GOTO60180
60110 AS$=AS$+QT$+RO$:GOTO60180
60120 IFQF=1AND(SC>=128)THENSC=SC-128:GOTO60140
60130 IFSC>=128THENSC=SC-128:RF=1:AS$=AS$+RV$
60140 IFSC<32ORSC>95THENAS=SC+64:GOTO60170
60150 IFSC>31ANDSC<64THENAS=SC:GOTO60170
60160 IFSC>63ANDSC<96THENAS=SC+32:GOTO60170
60170 AS$=AS$+CHR$(AS)
60180 IFRF=1THENAS$=AS$+RO$:RF=0
60190 NEXT RO
60200 IFQF=0THENPRINT#4,PO$"20"AS$:GOTO60220
60210 @PRINT#4,PO$"20"AS$QT$
60220 NEXT CL
60230 CLOSE4
60240 RETURN
```

Drean commodore

AGENTE AUTORIZADO



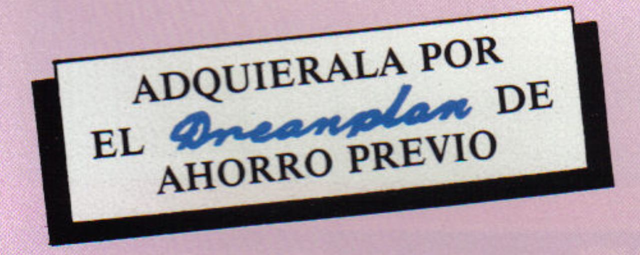



ENVIOS
AL INTERIOR

ASESORAMIENTO
GRATUITO A
ESCUELAS E
INSTITUTOS

Drean
commodore
C 16

Drean
commodore
C 64




No indicamos precios por teléfono

DISPONEMOS DE UN AMPLIO STOCK
DE SOFTWARE ORIGINAL C/GARANTIA
JOYSTICKS - BIBLIOGRAFIA - DISKETTES
INTERFACES - ACCESORIOS - GRABADORES
DISKETTERAS - IMPRESORAS Y DATASETE

AV. CORRIENTES 2198, ESQUINA URIBURU

**"LA ESQUINA DE LA COMPUTACION"**

TEL.: 46-2529/7877